University of Ioannina

School of Sciences, Physics Department

Development of Triggering Systems and Searches for Signatures from Theories beyond the Standard Model with the CMS detector at the LHC

Panos Katsoulis

supervised by
Dr. Costas Foudas

February 1, 2019

Abstract

The document in hand is separated into six main chapters including an Epilogue. In short, the first chapter describes theoretically the Standard Model of Particle Physics and also presents a few details about Supersymmetry (one of the most famous mathematical frameworks which extents the Standard Model). The second chapter describes the accelerating complex and the CMS experiment at CERN. Also, in the same chapter a short but inclusive presentation is performed about the Level-1 Trigger system, which is used by the CMS to record data. The following two chapters are dedicated to one of the subsystems of the triggering system, the ”Barrel Muon Track Finder” (BMTF). The third chapter presents the algorithms which are used by the BMTF and also describes the online software that is used for controlling and monitoring the system on the fly. In chapter four there is an extended description about the rest of the software which has been developed for supporting the system, but is used for different purposes than controlling and monitoring. In the last main chapter is discussed a study about using a newly included trigger in the CMS experiment, in the context of an analysis related to the search of Supersymmetric signatures. This trigger requires at least one muon to collect the event. Apart from these six main chapters, appendices exist covering complementary needs of this thesis.

Η παρόν εργασία είναι χωρισμένη σε έξι κεφάλαια συμπεριλαμβάνοντας τον Επίλογο. Επιγραμματικά, στο πρώτο κεφάλαιο περιγράφεται θεωρητικά το Καθιερωμένο Πρότυπο Σωματιδιακής Φυσικής και αναφέρονται λίγα πράγματα για την Υπερσυμμετρία (ένα από τα πιο περίφημα μοντέλα επέκτασης του καθιερωμένου προτύπου). Στο δεύτερο περιγράφεται το σύστημα επιτάχυνσης σωματιδίων και το πείραμα CMS στο CERN. Επίσης στο ίδιο κεφάλαιο γίνεται μία συνοπτική αλλά ολοκληρωμένη παρουσίαση του αρχικού συστήματος σκανδαλισμού (Level-1 Trigger) που χρησιμοποιείται από το πείραμα για να συλλεχθούν τα δεδομένα. Τα επόμενα δύο κεφάλαια εξειδικεύονται περισσότερο σε ένα από τα υποσυστήματα του συστήματος σκανδαλισμού, το σύστημα “Barrel Muon Track Finder” (BMTF). Το τρίτο κεφάλαιο παρουσιάζει τους αλγορίθμους που χρησιμοποιούνται από το BMTF και επίσης γίνεται μία περιγραφή του διαδικτυακού λογισμικού (online software) που χρησιμοποιείται για τον έλεγχο και την παρακολούθηση του συστήματος. Στο τέταρτο κεφάλαιο περιγράφεται εκτενέστερα το λογισμικό που έχει αναπτυχθεί για την υποστήριξή του συστήματος, το οποίο όμως χρησιμοποιείται για σκοπούς πέρα από τον έλεγχο και την παρακολούθηση. Θέμα του τελευταίου κεφαλαίου είναι η περιγραφή μιας μελέτης σχετικά με τη δυνατότητα χρήσης ενός πρόσφατα εισαχθέντος συστήματος σκανδαλισμού στο πείραμα CMS, από μία ανάλυση που ασχολείται με την αναζήτηση Υπερσυμμετρικών υπογραφών. Ο σκανδαλιστής στοχεύει γεγονότα με τουλάχιστον ένα μιόνιο. Υπάρχουν και πρόσθετα κεφάλαια τα οποία αναλύουν θέματα συμπληρωματικού χαρακτήρα.

Contents

Chapter 1

Particle Physics, The Standard Model and Beyond

1.1 Introduction

Particle physics is called the branch of science which studies the most fundamental blocks of the physical matter and also their interactions. Scientists who are involved in this subject investigate the different scenarios which describe what may exist beyond the known territories of what we call "The Standard Model of Particle Physics". This model, along with Maxwell's electromagnetism, is one of the 2 most successful theoretical scenarios in all the history of the physical science. It predicts correctly the existence and behavior of those fundamental "building blocks" up to an incredible agreement with the experimental test.

1.2 Quantum Field Theory, The Foundations of the Standard Model

From the theoretical point of view, the mathematical framework for understanding and analysing the "particle zoo" which came up from the past experiments are given by the Quantum Field Theory (QFT). In these theories the particles are manifestations of the disturbance of operator fields (for example $\Phi_i(x^\mu)$ that is a complex scalar, it could be a charged pion). The fields are quantized meaning that we have imposed on these mathematical structures commutation or anti-commutation relations (for bosons and fermions respectively) with their momentum ($\frac{\partial \Phi_i(x^\mu)}{\partial x_0} = \Pi_i(x^\mu)$).

$$\begin{aligned}
[\Phi_i(x^\mu), \Pi_j(y^\mu)]_{x^0=y^0} &= i\delta_{ij}\delta(\vec{x}-\vec{y}) \\
[\Phi_i(x^\mu), \Phi_j(y^\mu)]_{x^0=y^0} &= 0 \\
[\Pi_i(x^\mu), \Pi_j(y^\mu)]_{x^0=y^0} &= 0
\end{aligned} \tag{1.1}$$

The above expression, in the language of QFT, is necessary if someone wants to move from the concept of fields (that is used in this and the following sections) to the concept of particles which is the main point of interest for Particle Physics. The fields can be expressed via a Fourier expansion using the conjugate (to space x^μ) momentum space p^μ.

$$\Phi_i(x^\mu) = \int \widetilde{dp}\Big(a_i(\vec{p})e^{-ip^\mu x_\mu} + b_i^\dagger(\vec{p})e^{ip^\mu x_\mu}\Big) \tag{1.2}$$

Using this relation firstly we can write down the momentum of the field (Π_i). Consequently we are able to use these expressions together to express the Hamiltonian ($\mathcal{H} = E$) and the 3-momentum ($\vec{P}$) of each case (interaction) in terms of the operators $a_i(\vec{p})$ and $b_i(\vec{p})$ which are objects under the influence of the commutation (or anti-commutation) relations because of equation 1.1. These together form the 4-momentum operator $\mathscr{P}^\mu = (E, \vec{P})$.

Finally, in turns out that there exists a Hilbert space which is defined by the eigestates $|k_i\rangle$ of the momentum operator $\mathscr{P}^\mu$ (the Fock space). These eigenstates correspond to the fields Φ_i included in each scenario and are the representatives of the particles with momentum and energy that corresponds to the 4-momentum operator with which they are related by the eigenstate equation (1.3).

$$\mathscr{P}^\mu |k_i\rangle = P^\mu |k_i\rangle \tag{1.3}$$

Each eigenstate $|k_i\rangle$ can be expressed with respect to the vacuum eigenstate $|0\rangle$ which is the state of "no-particles". It is supposed to hold "infinite" energy which can be subtracted out for resulting to something able to be measured by an experiment (an energy difference). This is the first of a series of infinities that Quantum Theories suffer and is the easiest to cure. The reason for the infinity is that the corresponding integral for the energy (using $\mathcal{H}(\Phi_i, \Pi_i)$) diverges, so every theoretical energy calculation includes an "infinite". One can calculate the required energy differences, which the experiments are able to observe, by subtracting this divergence and defining the energy of the state $|0\rangle$ to be zero because the gravity is not present. Last but not least, in QFT the operators $a_i(\vec{k})$ and $b_i(\vec{k})$ behave as raising and lowering operators and can be used to "create" or "annihilate" a state of their corresponding momentum $\vec{k}$. Considering the above all together we can conclude how a particle is represented in the QFT framework, it's the following equations.

$$\begin{aligned} |k_i\rangle &= a_i^\dagger(\vec{k}_i)\,|0\rangle \\ |0\rangle &= a_i(\vec{k}_i)\,|k_i\rangle \end{aligned} \tag{1.4}$$

By applying once the raising operator $a_i^\dagger$ on the zero-particle state we conclude with a new particle, by applying once the lowering operator a_i on an one-particle state we conclude with zero particles.

Each particle has its characteristics like the charge, the colour etc. This information is also represented in QFT using similar (to $\mathscr{P}^\mu$) operators defined by their corresponding fields in the same fashion. The charges, along their corresponding currents of an interaction, form the 4-vector $\mathscr{J}^\mu = (Q_i, \vec{J}_i)$ which behaves like $\mathscr{P}^\mu$ in the Hilbert space of eigenstates $|k_i\rangle$. The eigenstate equation below gives the charge of the particle represented by the state $|k_i\rangle$ (or whatever the charge is related to in the interaction in question).

$$Q_i\,|k_i\rangle = q_i\,|k_i\rangle \tag{1.5}$$

In particle physics we take into account 3 fundamental forces, the Electromagnetism, the Weak and the Strong force. According to the Standard Model of Particle Physics these forces originate from the Electroweak and the Strong interactions. In QFT each interaction is formulated using the collection of fields related to the particles that are known to involve in the interaction and at least one more field which has the role of the mediator for the interaction in hand. A Lagrangian needs to be formed using these elements and because of the nature of the particle physics experiments it must be Lorentz invariant since they are taking place in high energies reaching the relativistic limit (without this to be a rare case scenario). More details about Quantum Field Theory can be found here [1], or in similar books.

1.3 The Standard Model of Particle Physics

Leptons and quarks are known to participate both in the Electroweak interactions, while Strong involve only quarks. The Lagrangian[1] for the model is separated into 3 parts.

$$\mathcal{L} = \mathcal{L}_{leptons}(EW) + \mathcal{L}_{quarks}(EW, Strong) + \mathcal{L}_{higgs}$$

[1] A different approach than what is selected here, which uses more the concepts of Group Theory, can be found in [2].

The required bosons which mediate the interactions are introduced by requiring the Standard Model Lagrangian to be invariant under local gauge transformations. This way the covariant derivatives for leptonic and quark fields are introduced. Leptons are divided into 3 generations (e and ν_e, μ and ν_μ, τ and ν_τ) and have their own part of the Lagrangian. The same happens for quarks too. They are divided into 3 generations (u and d, c and s, t and b). Mass terms for these fields are not gauge invariant but fermion masses can be introduced in a gauge invariant way via the Higgs Mechanism and Yukawa couplings.

1.3.1 Leptons

Leptons are one-half spin particles and according to QFT they can be represented by Weyl spinors (as well as Dirac and Majorana too). For the Standard Model the left-handed Weyl spinors for each generation can form a $SU(2)$ doublet while the remaining right-handed spinor is the $SU(2)$ singlet. The group of symmetry for the leptonic part of the Lagrangian ($\mathcal{L}_{leptons}$) is $SU(2) \times U(1)$ and the representation of the fields is $(2, \frac{-1}{2}) \bigoplus (1, +1)$[2]. For the leptonic part of the Lagrangian one can begin by the kinetic term and the Yukawa potential using the Higgs field Φ in the representation $(2, \frac{-1}{2})$. In what follows the index "g" indicates the different generations of leptons, "L" means left-handed field and "R" right-handed.

$$\mathcal{L}_{leptons} = i\overline{L}_g \gamma^\mu D_\mu L_g + i\overline{R}_g \gamma^\mu D_\mu R_g + (-y\overline{L}_g \Phi R_g + h.c.) \tag{1.6}$$

For concluding to terms which describe known interactions and masses firstly is necessary to substitute into the equation 1.6 the expression of the covariant derivatives. (Omitted "g" for 1.7)

$$\begin{gathered} D_\mu L = \partial_\mu L - ig_1(A^i_\mu \tau^i) \cdot L - ig_2\left(\frac{-1}{2}\right) B_\mu L \\ D_\mu R = \partial_\mu R - ig_2(+1) B_\mu R \end{gathered} \tag{1.7}$$

Next, higgs' doublet must be shifted by the non-zero vacuum expectation value V_0. This way a suitable minimum of higgs' potential is selected (see last subsection).

$$\Phi = \begin{pmatrix} \phi_1(x^\mu) \\ \phi_2(x^\mu) \end{pmatrix} \longrightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} V_0 + H(x^\mu) \\ 0 \end{pmatrix} \tag{1.8}$$

[2] This is a group theory notation. Indicates the representations of the fields (written inside the parentheses). Each slot of a parenthesis correspond to one of the underying symmetries of the model $(SU(2), U_Y(1))$. The direct sum indicates how these fields are combined.

In the end a change from $SU(2)$ doublets to Dirac and left-handed Majorana fields is comfort. One must keep in mind that a Dirac field is defined as a structure which holds the particle upper and the antiparticle lower. For the Majorana field the definition is quite similar if and only if we handle a particle which is its own antiparticle (charge zero, differently no difference in their quantum numbers).

$$\begin{aligned} \text{for electrons: } \Psi_{dirac} &= \begin{pmatrix} l \\ l^{\dagger} \end{pmatrix} \text{ e.g.} \begin{pmatrix} e^{-} \\ e^{+} \end{pmatrix} \\ \text{for their netrinos: } \Psi_{majorana} &= \begin{pmatrix} \nu \\ \nu^{\dagger} \end{pmatrix} \text{ e.g.} \begin{pmatrix} \nu_e \\ \overline{\nu}_e \end{pmatrix} \end{aligned} \tag{1.9}$$

The left-handed part of Majorana fields can be taken by acting on $\Psi_{majorana}$ with the operator $P_L = \frac{1}{2}(1 - \gamma_5)$, this means $L_g^{majorana} = P_L \Psi_g^{majorana}$.

Concluding, after expressing the gauges A_μ^i and B_μ using the fields $\mathcal{A}_\mu$, $W_\mu^\pm$ and Z_μ (details exist in next subsection) one can identify mass and interaction terms that are observed by experiments. ($\ell_g = \Psi_g^{dirac}$, $\ell'_g = P_L \Psi_g^{dirac}$ and $\nu_g = L_g^{majorana}$)

$$\begin{aligned} \mathcal{L}_{leptons} = & \mathcal{L}_{kin} + \frac{g_1}{\sqrt{2}} W_\mu^+ \overline{\nu}_g \gamma^\mu \ell'_g + \frac{g_1}{\sqrt{2}} W_\mu^- \overline{\ell'}_g \gamma^\mu \nu_g \\ & + \frac{e \cdot Z_\mu}{2} \left(\frac{\overline{\nu}_g \gamma^\mu \nu_g - \overline{\ell'}_g \gamma^\mu \ell'_g + 2 sin_w^2 \theta \cdot \bar{\ell}_g \gamma^\mu \ell_g}{sin_w \theta \cdot cos_w \theta} \right) \\ & - e \mathcal{A}_\mu \bar{\ell}_g \gamma^\mu \ell_g - \frac{y}{\sqrt{2}} (V_0 + H) \bar{\ell}_g \ell_g \end{aligned} \tag{1.10}$$

The equation 1.10 is a Lagrangian which describes the dynamics of the Electroweak interactions between leptons and bosons. For each interaction term, there is a current that can be extracted by the term in the Lagrangian. Integrating over the space one can conclude to the charge operators that are described in the QFT section. While the Lagrangian is constructed at the point when the group gauges expressed in terms of the bosons Z, $W^\pm$ and γ one can extract for the generators the relation $Q = \tau^3 + Y$. Also, because of the representations stated in the beginning for $SU(2)$ and $U(1)$-hypercharge, the below eigenstate equations hold.

$$\begin{aligned} \tau^3 L_g^2 &= \frac{+1}{2} L_g^2, \quad \tau^3 L_g^1 = \frac{-1}{2} L_g^1, \quad \tau^3 R_g = 0 \\ Y L_g^2 &= \frac{-1}{2} L_g^2, \quad Y L_g^1 = \frac{-1}{2} L_g^1, \quad Y R_g = +1 R_g \end{aligned} \tag{1.11}$$

Hence, the extracted equation results into $Q(\ell_L) = -1$, $Q(\nu) = 0$ and $Q(\ell_R) = +1$ as eigenvalues for the related eigenstate equations, similar to the equation 1.5.

1.3.2 Quarks

Quarks, like leptons are one-half spin particles and can also be represented by Weyl spinors. Left-handed Weyl spinors of different generations can also form $SU(2)$ doublets while the remaining 2 right-handed spinor are $SU(2)$ singlets and $SU(3)$ triplets in the adjoint representation. The symmetry group of the Lagrangian ($\mathcal{L}_{quarks}$) is $SU(3) \times SU(2) \times U(1)$ and the representation of the fields is $(3, 2, \frac{1}{6}) \bigoplus (\bar{3}, 1, \frac{-2}{3}) \bigoplus (\bar{3}, 1, \frac{+1}{3})$[3].

For the Electroweak interactions of quarks one can begin by the kinetic term and the Yukawa potential using the Higgs field Φ in the representation $(1, 2, \frac{-1}{2})$. Then following the same procedure like for leptons one can conclude into a Lagrangian that describes the dynamics between quarks, the gauge bosons and the higgs. In what follows the index "g" indicates the different generations of quarks, "L" means left-handed field and "R" right-handed.

$$\mathcal{L}_{quarks} = i\overline{L}_g\gamma^\mu D_\mu L_g + i\overline{R^i}_g\gamma^\mu D_\mu R^i_g + (-y'_{gg'}\overline{L}_g\Phi R^1_{g'} - y''_{gg'}\overline{L}_g\Phi R^2_{g'} + h.c.) \tag{1.12}$$

The procedure is the same as for leptons. Firstly one substitutes into the equation 1.12 the expression of the covariant derivatives. (L here holds 2 indices one for $SU(2)$ like before and one for $SU(3)$, omitted "g" and $SU(2)$-index for 1.13)

$$\begin{gathered} D_\mu L_a = \partial_\mu L_a - ig_0(A^j_\mu \tau^j_{su3})_{ab}L_b - ig_1(A^j_\mu \tau^j) \cdot L_a - ig_2\Big(\frac{+1}{6}\Big)B_\mu L_a \\ D_\mu R^i_a = \partial_\mu R^i_a - ig_0(A^j_\mu \tau^j_{su3})_{ab}R^i_b - ig_2(Y_i)B_\mu R^i_a \end{gathered} \tag{1.13}$$

Same as before, the higgs' doublet must be shifted and the suitable minimum of higgs potential must be selected (see last subsection).

$$\Phi \longrightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} V_0 + H(x^\mu) \\ 0 \end{pmatrix} \tag{1.14}$$

Next, changing from $SU(2)$ doublets to Dirac fields is required for the quarks too.

$$\text{for quarks: } \Psi_{dirac,a} = \begin{pmatrix} q_a \\ q^\dagger_a \end{pmatrix} \text{ e.g.} \begin{pmatrix} d_a \\ d^\dagger_a \end{pmatrix} \tag{1.15}$$

[3] $(SU(3), SU(2), U_Y(1))$

The left-handed fields can be taken by acting on Ψ_{dirac} with the operator $P_L = \frac{1}{2}(1-\gamma_5)$, this means $L_g^{dirac} = P_L \Psi_g^{dirac}$.

As last step of the procedure the gauges A_μ^{ai} ($SU(2)$, $a = 1,2,3$) and B_μ are expressed by the fields $\mathcal{A}_\mu$, $W_\mu^\pm$ and Z_μ (see last subsection), and from that step one keeps the relation $Q_i = \tau^3 + Y_i$. ($q_{ig}^a = \Psi_{ig}^{dirac,a}$, $q_{ig}'^a = P_L \Psi_{ig}^{dirac,a}$, $q_{1g}^a = (u^a, c^a, t^a)$, $q_{2g}^a = (d^a, s^a, b^a)$ and $i = 1,2$)

$$\begin{aligned}\mathcal{L}_{quarks} =& \mathcal{L}_{kin} + \mathcal{L}_{strong} + \frac{g_1}{\sqrt{2}} W_\mu^+ \overline{q'}_{1g}^a \gamma^\mu q_{2g}'^a + \frac{g_1}{\sqrt{2}} W_\mu^- \overline{q'}_{2g}^a \gamma^\mu q_{1g}'^a \\ &+ \frac{e \cdot Z_\mu}{2} \left[\frac{\overline{q'}_{1g}^a \gamma^\mu q_{1g}'^a - \overline{q'}_{2g}^a \gamma^\mu q_{2g}'^a + 2sin_w^2\theta \cdot Q_i \overline{q}_{ig}^a \gamma^\mu q_{ig}^a}{sin_w\theta \cdot cos_w\theta} \right] \\ &- e\mathcal{A}_\mu Q_i \overline{q}_{ig}^a \gamma^\mu q_{ig}^a - \frac{1}{\sqrt{2}}(V_0 + H)(y_{gg'}' \overline{q}_{1g}^a q_{1g'}^a + y_{gg'}'' \overline{q}_{2g}^a q_{2g'}^a)\end{aligned} \tag{1.16}$$

This is the final equation for the quarks' dynamics under the Electroweak interactions. In the above Lagrangian is used the symbol Q_i, these are the eigenvalues that can be calculated by making use of the extracted relation $Q_i = \tau^3 + Y_i$. One can result in $Q_1(q_g) = \pm\frac{2}{3}$ and $Q_2(q_g) = \mp\frac{1}{3}$ if use the information for $U(1)$-hypercharges and $SU(2)$-charges given by the fields' representations in the beginning of this subsection.

Focusing on the currents' terms coupling with $W_\mu^\pm$ of $\mathcal{L}_{quarks}$ (the last 2 terms in the first line of eq. 1.16), one can formulate it into matrix expressions like below. Also, a unitary transformation of the left-handed quark fields leads to the Cabibbo-Kobayasi-Maskawa (CKM) matrix. The transformations are $q_{1g}'^a \longrightarrow U_{1gg'} q_{1g'}'^a$ and $q_{2g}'^a \longrightarrow U_{2gg''} q_{2g''}'^a$.

$$\begin{aligned}\overline{q'}_{1g}^a \gamma^\mu q_{2g}'^a &\longrightarrow (\overline{q'}_{1g'}^a U_{1g'g}^*) \gamma^\mu (U_{2gg''} q_{2g''}'^a) = \overline{q'}_{1g'}^a [U_1^* U_2]_{g'g''} \gamma^\mu q_{2g''}'^a \\ \overline{q'}_{2g}^a \gamma^\mu q_{1g}'^a &\longrightarrow (\overline{q''}_{2g''}^a U_{2g''g}^*) \gamma^\mu (U_{1gg'} q_{1g'}'^a) = \overline{q''}_{2g'}^a [U_2^* U_1]_{g''g'} \gamma^\mu q_{1g'}'^a\end{aligned} \tag{1.17}$$

One can extract from the above expressions the CKM-matrix, it follows.

$$V_{gg'} = [U_1^* U_2]_{gg'} \tag{1.18}$$

In both equations 1.13, except the gauges that have been used to create the fields $\mathcal{A}_\mu, W_\mu^\pm$ and Z_μ^0, also exist 8 gauges that correspond to the $SU(3)$ symmetry and have been suppressed in the term $\mathcal{L}_{strong}$ of the Lagrangian. As the naming scheme of the term indicates, this is the part that describes the strong interactions. By substituting this last term of the covariant derivative into the quarks' kinematic term one would result in the following equation.

$$\mathcal{L}_{strong} = i\overline{q'}^{a}_{ig}\gamma^{\mu}\partial_{\mu}q'^{a}_{ig} + g_0\overline{q'}^{a}_{ig}\gamma_{\mu}(A^{j}_{\mu}\tau^{j}_{su3})^{ab}q'^{b}_{ig} - \frac{1}{4}G^{a}_{\mu\nu}G^{a\mu\nu} \tag{1.19}$$

The last term of 1.19 is the kinetic term for the gauges A^{aj}_{μ} ($SU(3)$, $a = 1,2,...8$) and it includes "self-interacting" terms. The form of it is illustrated bellow.

$$\begin{aligned}-\frac{1}{4}G^{a}_{\mu\nu}G^{a\mu\nu} = \mathcal{L}\left((\partial A)^2 \text{ terms}\right) + \frac{1}{4}f^{abc}f^{ade}A^{b}_{\mu}A^{c}_{\nu}A^{\mu d}A^{\nu e} \\ + \frac{i}{4}\left(f^{ade}(\partial_{\nu}A^{a}_{\mu} - \partial_{\mu}A^{a}_{\nu})A^{\mu d}A^{\nu e} + f^{bca}A^{b}_{\mu}A^{c}_{\nu}(\partial^{\nu}A^{\mu a} - \partial^{\mu}A^{\nu a})\right)\end{aligned} \tag{1.20}$$

So in the end one would conclude with the Lagrangian below in which the interacting terms of chromodynamics are now obvious.

$$\begin{aligned}\mathcal{L}_{strong} = \mathcal{L}_{kin} &+ g_0\overline{q'}^{a}_{ig}\gamma_{\mu}(A^{j}_{\mu}\tau^{j}_{su3})^{ab}q'^{b}_{ig} \\ &+ \frac{if^{abc}}{4}\left((\partial_{\nu}A^{a}_{\mu} - \partial_{\mu}A^{a}_{\nu})A^{\mu b}A^{\nu c} + A^{b}_{\mu}A^{c}_{\nu}(\partial^{\nu}A^{\mu a} - \partial^{\mu}A^{\nu a})\right) \\ &+ \frac{1}{4}f^{abc}f^{ade}A^{b}_{\mu}A^{c}_{\nu}A^{\mu d}A^{\nu e}\end{aligned} \tag{1.21}$$

In the first line, except the kinematic terms which are suppressed, one could identify the interactions of the quark fields with the $SU(3)$ gauge fields (the gluons). In the second and third lines there are terms which describe the "self-interacting" capability of the $SU(3)$ gauges. This is because such non-abelian (Yang-Mills) theories always include an extra term in "their" strength tensor which gives raise to terms like those in the last 2 lines of the Lagrangian 1.21. The same happens for the $SU(2)$ symmetry (as non-abelian) and is illustrated in the next subsection in a different aspect because the gauges of $SU(2)$ are mixed along with the gauge of $U(1)$-hypercharge to form $W^{\pm}_{\mu}$ and Z^{0}_{μ} fields. For the strong interaction the charge operators Q_a which act on the particle eigenstates $\left|q_{ag}\right\rangle$ have eigenvalues the $SU(3)$ charge (or differently the charge of strong interactions which is the color of quarks).

1.3.3 The Higgs Mechanism and the Gauge Fields

In the previous two subsections the interactions have been introduced into the formulation by using the gauge fields A_{μ} and $W^{\pm}_{\mu}, Z^{0}_{\mu}$. These fields have their own kinetic terms and some of them are observed massive, but a mass term is not allowed to be included in the Lagrangian because these terms are violating gauge invariance. To introduce the correct mass terms for the gauges one has to rely on the "Higgs Mechanism". The higgs field is a complex scalar $SU(2)$ spinor and is in the representation $(1, 2, \frac{-1}{2})$, a singlet of $SU(3)$ with which doesn't interact (gluons do not acquire mass terms). One can begin with the following Lagrangian which respects the symmetries of $\mathcal{L}_{leptons} + \mathcal{L}_{quarks}$.

$$\mathcal{L}_{higgs} = -(D^{\mu}\Phi)^{\dagger}(D_{\mu}\Phi) - \frac{\lambda}{4}\Big(\Phi^{\dagger}\Phi - \frac{V_0^2}{2}\Big)^2 \tag{1.22}$$

The covariant derivative here must be substituted by the following expression with the gauge fields. Their related field strength tensors $F^i_{\mu\nu}$ and $B_{\mu\nu}$ are also required to be included in $\mathcal{L}_{higgs}$ for their kinematic terms.

$$D_{\mu}\Phi = \partial_{\mu}\Phi - ig_1(A^{\alpha}_{\mu}\tau^{\alpha})\Phi - ig_2\Big(\frac{-1}{2}\Big)B_{\mu}\Phi \tag{1.23}$$

Easily can be shown that the potential of the above Lagrangian has infinite number of minima for $|\Phi| = \frac{V_0}{\sqrt{2}}$, and means that the vacuum expectation value (VEV) for Φ is non-zero. We are free to select this VEV and express the Lagrangian in terms of one of these minima. The most convenient configuration of Φ_{min} is when one makes a global gauge transformation and brings the VEV entirely to one component of the spinor, and furthermore by using the Unitary gauge 1 out of 2 remaining degrees of freedom can be fixed at $\chi(x) = 0$.

$$\Phi(x^{\mu}) \longrightarrow \frac{1}{\sqrt{2}}\begin{pmatrix} V_0 + H(x^{\mu}) \\ 0 \end{pmatrix} \tag{1.24}$$

This is where this substitution comes from and used in the Yukawa potentials with leptons and quarks, the Higgs field that gives mass to fermions is the same field that gives mass to the gauge fields as well. So rewriting the potential in terms of the shifted higgs field, and also making use of the expression for the covariant derivative, one would have mass terms only for some of the mediators and the higgs field (H). But the field Φ initially was containing 4 degrees of freedom and after this procedure only one of them is massive (we may use the above expression for the higgs field but this is only for the potential of $\mathcal{L}_{higgs}$, all real 4 fields of Φ still have kinematic terms because the transformations described above are actually a phase transition and a spinor rotation). What actually happens here is that the initial $SU(2) \times U(1)$ has broken into a new $U(1)$, the electromagnetism. When global symmetries are being broken down Goldstone bosons appear, and in this case there are 3 goldstone bosons, but this was a gauge symmetry and the Goldstone bosons "consumed" by $W^{\pm}$ and Z^0 and in turn they became massive.

Before someone obtain the final form of $\mathcal{L}_{higgs}$ 2 more changes are necessary. Firstly, change the gauge fields introduced by the covariant derivatives to $\mathcal{A}_\mu, Z^0_\mu$ and $W^\pm_\mu$ using the equations below, and define the Weinberg angle as $\theta_w = \arctan\frac{g_2}{g_1}$.

$$\begin{aligned}\mathcal{A}_\mu &= sin\theta_w A^3_\mu + cos\theta_w B_\mu \\ Z^0_\mu &= cos\theta_w A^3_\mu - sin\theta_w B_\mu \\ W^\pm_\mu &= \frac{1}{\sqrt{2}}(A^1_\mu \mp iA^2_\mu)\end{aligned} \tag{1.25}$$

And secondly, define the covariant derivative for $W^\pm_\mu$ and express the field strength tensors $F^i_{\mu\nu}, B_{\mu\nu}$ using $F_{\mu\nu}$ and $Z_{\mu\nu}$. The useful relations follow.

$$\begin{aligned}D_\mu W^\pm_\nu &= (\partial_\mu - ig_1 \sin\theta_w \mathcal{A}_\mu - ig_1 \cos\theta_w Z^0_\mu) W^\pm_\nu \\ \frac{1}{\sqrt{2}}(F^1_{\mu\nu} - iF^2_{\mu\nu}) &= D_\mu W^+_\nu - D_\nu W^-_\mu \\ \frac{1}{\sqrt{2}}(F^1_{\mu\nu} + iF^2_{\mu\nu}) &= D^\dagger_\mu W^+_\nu - D^\dagger_\nu W^-_\mu \\ F^3_{\mu\nu} &= \partial_\mu A^3_\nu - \partial_\nu A^3_\mu - ig_1(W^+_\mu W^-_\nu - W^+_\nu W^-_\mu) \\ B_{\mu\nu} &= \cos\theta_w F_{\mu\nu} - \sin\theta_w Z_{\mu\nu}\end{aligned} \tag{1.26}$$

Using all these substitutions, and after the Higgs mechanism has taken place one can conclude into this Lagrangian.

$$\begin{aligned}\mathcal{L}_{higgs} = &-\frac{1}{4}F_{\mu\nu}F^{\mu\nu} - \frac{1}{4}Z_{\mu\nu}Z^{\mu\nu} - D^{\dagger\mu}W^{-\nu}D_\mu W^+_\nu + D^{\dagger\mu}W^{-\nu}D_\nu W^+_\mu \\ &+ ie(F^{\mu\nu} + \cot\theta_w Z^{\mu\nu})W^+_\mu W^-_\nu \\ &- \frac{e^2}{\sin^2\theta_w}(W^{+\mu}W^-_\mu W^{+\nu}W^-_\nu - W^{+\mu}W^+_\mu W^{-\nu}W^-_\nu) \\ &- (M^2_W W^{+\mu}W^+_\mu + \frac{1}{2}M^2_Z Z^\mu Z_\mu)(1 + V_0^{-1}H)^2 \\ &- \frac{1}{2}(\partial^\mu H \partial_\mu H + m^2_H H^2 + \frac{m^2_H}{V_0}H^3 + \frac{m^2_H}{4V_0^2}H^4)\end{aligned} \tag{1.27}$$

This is the final equation for the higgs part of the Lagrangian. Someone could identify these terms one by one. Firstly, in the first line the kinematic terms for $\mathcal{A}_\mu, Z^0_\mu$ and $W^\pm_\mu$. The second line describes the coupling of $\mathcal{A}_\mu$ and Z^0_μ to $W^\pm_\mu$ and the third line are the "self-coupling" terms for $W^\pm_\mu$. The forth line contains the mass terms for $W^\pm_\mu$ and Z^0_μ, and also their couplings to the higgs field. In the last line one can identify only higgs' terms, its kinetic, mass and self-coupling parts. The symbols e, M_W, M_Z and m_H are defined as the calculation towards eq. (1.24) proceeds.

$$\begin{gathered} e = g_1 \sin(\theta_w) \\ M_W = \frac{g_1 V_0}{2} \quad \text{and} \quad M_Z = \frac{M_W}{\cos\theta_w} \\ m_H = \sqrt{\frac{\lambda V_0^2}{2}} \end{gathered} \tag{1.28}$$

As conclusion in this subsection (which also completes the Standard Model of Particle Physics) let's refer to the constants that have been encountered in the model. They are very important because they specify it completely. The 3 gauge couplings (g_0, g_1, g_2) for each interaction. For each Yukawa diagonal matrix, there are 3, the 3 entries of the primary diagonal. For the CKM matrix one can prove that it has 4 independent real entries. The scalar quadratic coupling (λ) and the scalar mass squared included in the higgs potential. The list can be extended if someone consider experimental results too. The masses, 3 for leptons, 6 for quarks, 2 for gauges and 1 for higgs, they are not predicted theoretically but have been observed.

1.4 Beyond the Standard Model

The model described in the previous sections even if it has very good agreement with the experiment up to now, it has also some known problems and it cannot be considered as the Theory of Everything. Some of these problems and questions are listed here.

- Absence of Gravity
  There is no any known procedure, up to now, which can lead a QFT for gravity towards Einstein's General Relativity.

- Hierarchy Problem
  The theoretical masses and coupling constants if don't be fine-tuned are non-consistent with observations.

- Generations of Fermions
  There is no explanation why exist only 3 generations of fermions, while there is no theoretical limitation under the Standard Model.

- Neutrino Masses
  Mass terms for neutrinos don't exist under the Standard Model.

- Matter/Antimatter Asymmetry
  There is no explanation why the known matter is created by particles and no by antiparticles.

1.4.1 Supersymmetry

Towards a better understanding, scientists work on several theoretical scenarios to extend the Standard Model of Particle Physics. The most popular scenario is called ”Supersymmetry” and proposes a new symmetry which postulates that for each fermion there exist a boson and vice versa. Supersymmetry is not a specific theory, it introduces a framework for many theoretical models which share common characteristics and concepts.

A Supersymmetric model is said to be formulated in the ”superspace”, this is an expanded structure of the spacetime and except x^{μ} it holds 2 more degrees of freedom a left-handed spinor θ and its right-handed complex conjugate θ^{*}. An other concept which also is extended is the conept of the field. A field in the superspace needs to be function also of the spinors and so it is. Fields like $\Phi = \Phi(x^{\mu}, \theta, \theta^{*})$ are called ”superfields”.

To construct a theoretical model, a Lagrangian is needed. In Supersymmetry Lagrangians consist of terms whose integral $\int d^4x$ are invariant under supersymmetry transformations. These terms are called F-terms $(W(\Phi)|_F)$ and D-terms $(\Phi^{\dagger}\Phi|_D)$. They are products of left-handed chiral and vector superfields respectively. The F-term is called ”Superpotential”. Supersymmetric Lagrangians are of the following form.

$$\mathcal{L} = \Phi^{\dagger}\Phi|_D + (W(\Phi)|_F + h.c.)$$

1.4.2 The Minimal Supersymmetric Standard Model

Here is described briefly the simplest extension of the Stadard Model, its abbreviation is ”MSSM”. The Lagrangian respects invariance of the gauge group $SU(3) \times SU(2) \times U(1)$ and the model consists of vector superfields for the bosons and their superparteners (gauginos), and chiral superfields for leptons and sleptons, quarks and squarks. All of them are in their known representations by the Standard Model. Also, there are 2 more chiral superfields for 2 copies of usual higgses with representation $(1, 2, \frac{-1}{2})$ and their sparteners the higgsinos with representation $(1, 2, \frac{+1}{2})$. The superpotential consists by Yukawa-like terms and is the following (for the superfields here are used the same symbols with those in SM, but all in capitals).

$$W = -y_{gg'}\varepsilon^{ij}H_iL_{jg}\bar{R}_{g'} - y'_{gg'}\varepsilon^{ij}H_iQ^a_{jg}\bar{D}^a_{g'} - y''_{gg'}\varepsilon^{ij}H_iQ^a_{jg}\bar{U}^a_{g'} - \mu\varepsilon^{ij}\bar{H}_iH_j \quad (1.29)$$

In the superpotential one could have included terms of the form $\varepsilon^{ij}\bar{H}_i L_j$ since they are not violating the gauge invariance. However these terms are not present in W because in MSSM the Lagrangian respects the R-parity[4] symmetry and the terms are not allowed.

The superpotential W is used to create F-terms for the Lagrangian using all its gauge invariant terms, but first one more superfield must be introduced. Supersymmetry must be able to break spontaneously because it's not an exact symmetry of the world we live into. Phenomenology requires the Supersymmetry Braking to be triggered by different fields than those introduced for MSSM until now. Effects from other superfields can be described by a new field $S = m_S^2 \theta_a \theta_a$ (D and F terms are defined as coefficients of $\theta\theta$ and $\theta\theta\theta^*\theta^*$ in superfield products). The complete Lagrangian for MSSM follows (the word "fields" under the sum symbol indicates leptons' and quarks' superfields only).

$$\begin{aligned} \mathcal{L}_{MSSM} = & \left(\frac{S^\dagger S}{m_M^2} \sum_{fields} \sum_{g,g'} C_{gg'} \Phi_g^\dagger \Phi_{g'} + C_H H^\dagger H + C_{\bar{H}} \bar{H}^\dagger \bar{H} \right) \Bigg|_D \\ & + \left\{ \frac{S}{m_M} \left(\sum_{\text{inv.terms}} C_i W_i|_F + \sum_{i=1,2,3} C_i' (A_{ia} A_{ia})|_F \right) + h.c. \right\} \Bigg|_F \end{aligned} \tag{1.30}$$

[4] The R-parity is a Z_2 symmetry. It is imposed to restore the baryon and lepton numbers conservation during the Supersymmetric interactions. The symmetry doesn't allow terms which describe interactions that violate these conservation laws.

Chapter 2

CERN,
Accelerators and the CMS

2.1 Introduction

The European Organisation for Nuclear Research (or in french Conseil Européen pour la Recherche Nucléaire - CERN) hosts some of the largest experiments in the world, targeting to push the human knowledge beyond the known limits. Scientists and engineers from different places on earth meet there and exchange ideas, as well as advanced knowledge and technology. Their goal is to make some of the most advanced apparatuses the human kind has ever constructed and operated, for scientific purposes. Inside CERN multiple accelerators and massive detectors (CMS, ATLAS, ALICE, LHCb) are functioning synchronized. Also very advanced computing is taking place there and network ideas and implementations are being tested and used. It is vital to be highlighted that this fundamental research returns very important benefits for all the society, even if people who aren't members of the CERN community don't realise it directly. The CERN facilities cover territories of 2 countries, France and Switzerland, near Geneva city.

2.2 Accelerators

The facilities host a complex system of several accelerators which creates and accelerates beams up to a center of mass energy of 14 TeV. Firstly, the LINAC machines which have as target to accelerate the substance in question to low energies and then to deliver them to the synchrotron complex. There are 4 of them until now at CERN, LINAC-1, 2, 3 and 4. LINAC-2 is the successor of the historical LINAC-1 (on 1978) and the main pre-accelerator for the Proton Physics project. It accelerates protons linearly up to 50 MeV (5 % mass gain). The LINAC-3 handles ions since 1994. It provides ion beams for the Heavy Ion Physics project like the Xenon-Xenon runs. The most modern linear accelerator, still under development, is the LINAC-4. It is scheduled to take over from LINAC-2 on 2020 as the main pre-accelerator of the LHC Chain. It will be capable to accelerate negative hydrogen up to 160 MeV and then to deliver the beam to the synchrotrons for further acceleration.

Moving towards the synchrotrons, 4 of them exist in the LHC Chain (the sequence of accelerators up to the Large Hadron Collider). The LINAC machines are designed to inject their beams to the Proton Synchrotron Booster (PSB). Four synchrotrons that are placed one on top of the other vertically are the constituents of the PSB which receives the beams from the LINACs. After accelerating them up to 1.4 GeV, the beams are being ejected, combined and transferred to the next synchrotron. The PSB was still under development on the 60s and received its first beams on 1972. Before the Booster was constructed the beams were being injected directly into the next synchrotron of the LHC Chain the Proton Synchrotron (PS). The PS was the first synchrotron of the CERN's complex and it has undergo through multiple development with its characteristics to be always on the cutting edge through the last decades. This synchrotron once receive the beam from PSB (for the usual Proton Physics case) accelerates it up to 25 GeV before handle it over to the next machine. Also, according to CERN's site which includes information about the accelerators, PS has accelerated other substances as well, alpha particles, oxygen, sulphur and also electrons for the pre LHC era, once LEP was still functioning.

The last two accelerating steps are the Super Proton Syncrotron (SPS) and the Large Hadron Collider (LHC). These 2 machines takes the substances at the GeV scale and deliver them to the four main experiments (CMS, ATLAS, ALICE and LHCb) at the TeV scale. The SPS turned-on in 1976 and after CERN picked-up the proton-antiproton challenge of those days SPS started operating in 1981 as a proton-antiproton collider. At that era the mayor highlights were the discoveries of W and Z bosons (1983), and also the direct discovery of CP-Violation (1999). After going through several upgrades, these days SPS is delivering high-luminosity proton beams to the LHC at 450 GeV. In order to have this capability, it was necessary changes to be done at most of its design characteristics to meet the technical requirements for the LHC to operate as designed. Through the years SPS accelerated successfully protons, antiprotons, electrons, positrons and also heavy-ions. This accelerator is considered as one of the most important for CERN's synchrotron complex and is planned, along with the entire complex, to be upgraded for the High Luminosity LHC (HL-LHC) era.

Last but not least the LHC. This is the most advanced accelerating machine the human race has ever constructed up to these days and started operating in 2008. The accelerator is hosted inside a 26.7 km ring-tunnel, at 50 to 175 m distance under the surface. It utilises superconducting magnets operating at temperature close to 0K (-273^oC) and several accelerating methods, to make beams of opposite direction to move with speed more than 99% of the speed of light in vacuum. (For the speed and the relativistic-γ of protons inside the LHC the Wikipedia site reports 0.999999990c and 6.930 respectively.) The designed luminosity of the LHC machine was $10^{34}cm^{-2}s^{-1}$ but $2 \times 10^{34}cm^{-2}s^{-1}$ achieved in 2017. Once the protons received from SPS, LHC is responsible to deliver to experiments beams of 7 GeV each with the above luminosity.

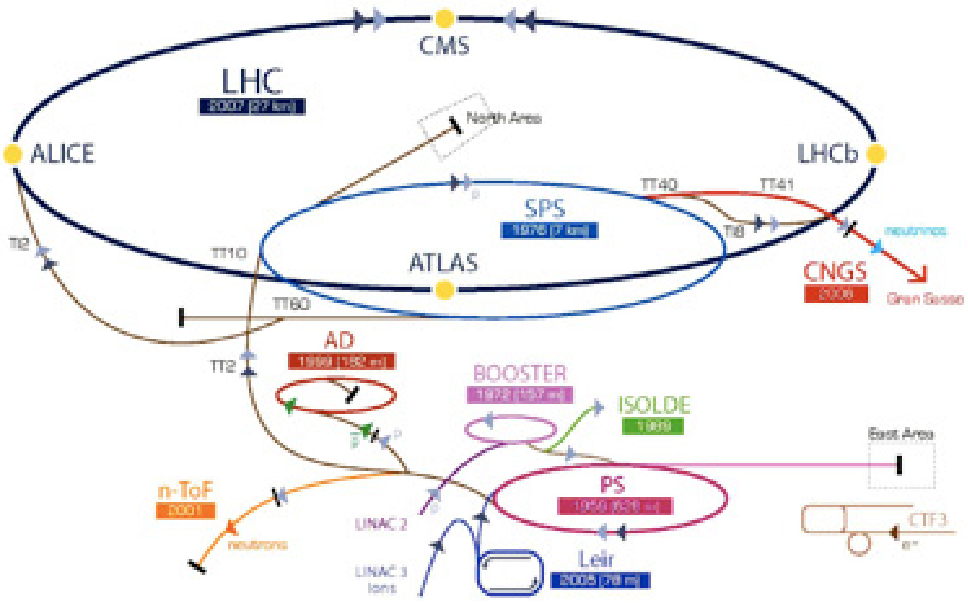


Figure 2.1: CERN Accelerator Complex

2.3 The CMS Detector

The CMS apparatus is designed as a general-purpose detector. Its main goal is to be capable to study proton-proton collisions at the designed energy and luminosity of the LHC. The CMS is consisted by several high technology sub-detectors, that each of them is designed to measure something different. Gathering together the information from every sub-detector one is capable to reconstruct what happened after the collisions are taking place. The subdetectors generate signals, when particles interact with them, which are being amplified and read-out by the front-end electronics. Namely, the layers of the detector are the Tracker, the Calorimeters and the Muon detectors. One important measurement is the transverse momentum of the particles. For this purpose a 3.8 Tesla magnet is installed between the Calorimeters and the Muon detectors.

The Silicon Tracker is the inner most layer of the CMS. Its main purpose is to track the position of the particles as they fly towards the Calorimeters. Each position measurement is accurate to $10\mu m$. Using these position measurements it's possible to have a momentum estimation with high precision. Tracker is consisted by 13 silicon layers at the area of the barrel, and 14 layers at the endcaps of the detector.

The CMS has 2 kinds of Calorimeters, the Electromagnetic Calorimeter [4] (ECAL) and the Hadronic Calorimeter [5] (HCAL). The ECAL is designed to measure the energy deposits, and positions, of particles that interact electromagnetically. This calorimeter is build by lead tungsten crystals. The material is ideal to stop high energy particles and once electrons or photons cross it scintillates (produces light). The front-end electronics are responsible for measuring and record the scintillating signals. ECAL is organized in 2 sections, the Barrel Calorimeter ($|\eta| < 1.48$) and the Endcap Calorimeter ($|\eta| > 1.48$). Part of the ECAL sub-detector is the preshower, it is installed in front of the Endcap and its main goal is to help the reconstruction distinguishing photons from pions. The preshower covers the area $1.65 < |\eta| < 2.61$

The HCAL is the next sub-detector and is installed after ECAL. Its purpose is to measure energy deposits and positions of hardons (compositions of quarks and gluons) and their produced showers. HCAL is organized into 3 parts, the Central section ($|\eta| < 3.0$) which includes Barrel and Endcap, the Forward section ($3.0 < |\eta| < 5.0$), and the Outer section which lies after the coil. The outer section exists because the amount of material needed for an effective measurement of a mean shower. This length is about 1 meter towards the traveling direction of the shower.

The Magnet of the CMS is of central importance to the experiment because its bending power is used to measure the momentum of the charged particles. The operating current while the CMS is collecting data is about 18.1 kA (magnetic field of 3.8T). This corresponds to stored magnetic energy of about 2.3 GJ. Special measures of safety have been taken for dumping the stored energy (via special dumping circuit) and limiting the magnetic field (using a return yoke right after the Muon subsystem).

After the coil and the outer hadronic calorimeter lies the Muon System

[3]. It consists of three detector technologies the Drift Tube Chambers (DTs, $|\eta| < 1.3$), the Cathode Strip Chambers (CSCs, $0.9 < |\eta| < 2.4$) and the Resistive Plate Chambers (RPCs, $|\eta| < 2.1$). In the Barrel section, where the magnetic field is weaker, DTs are the most efficient technology to use because of their compactness. In the Endcap the magnetic field is stronger and also the outgoing particles (muons and background) are more. Because RPCs have better time resolution of their measurements, they are more efficient to be used there. The RPCs have very fast response time and are used, both in the Barrel and in the Endcap, to record muon tracks with a very accurate bunch crossing assignment.

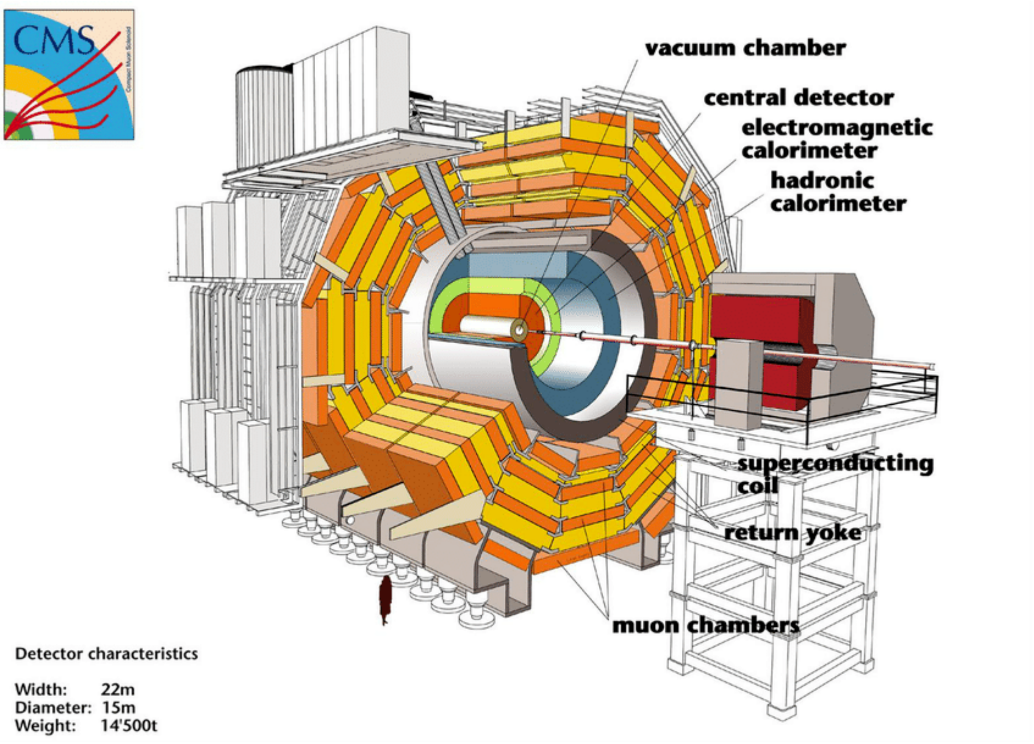


Figure 2.2: The CMS Detector

2.4 The CMS Level 1 Trigger

The L1 Trigger (L1T) is responsible to decide if the data of an event will be stored or not. Once collecting all the hits produced by the CMS sub-detectors L1T uses various algorithms, such as pattern recognition, to decide if the event in question is likely to include interesting physics. Interesting physics by the scope of the several analyses that take place under the aegis of the CMS experiment. The L1 Trigger [6] is a collection of very fast electronics (general-purpose triggers implemented on FPGAs) that are designed to filter the incoming data and deliver an output of 100kHz to the CMS offline computing clusters.

A L1 decision, positive to store data or negative, is designed to be delivered to the frontend electronics every 25ns. This is the amount of time between two sequential bunch crossings. When the incoming data from a bunch crossing occur they are preserved to frontend buffers while the L1 decision is being calculated. The target of the L1T system is to deliver the decision while keeping the operational deadtime negligible, otherwise the frontend buffer will be full and collision data will begin to be overwritten.

The Data Acquisition system of each sub-detector deliver their primitives to the L1 FPGAs. The figure below shows the architecture of the L1 Trigger system (Figure 2.3).

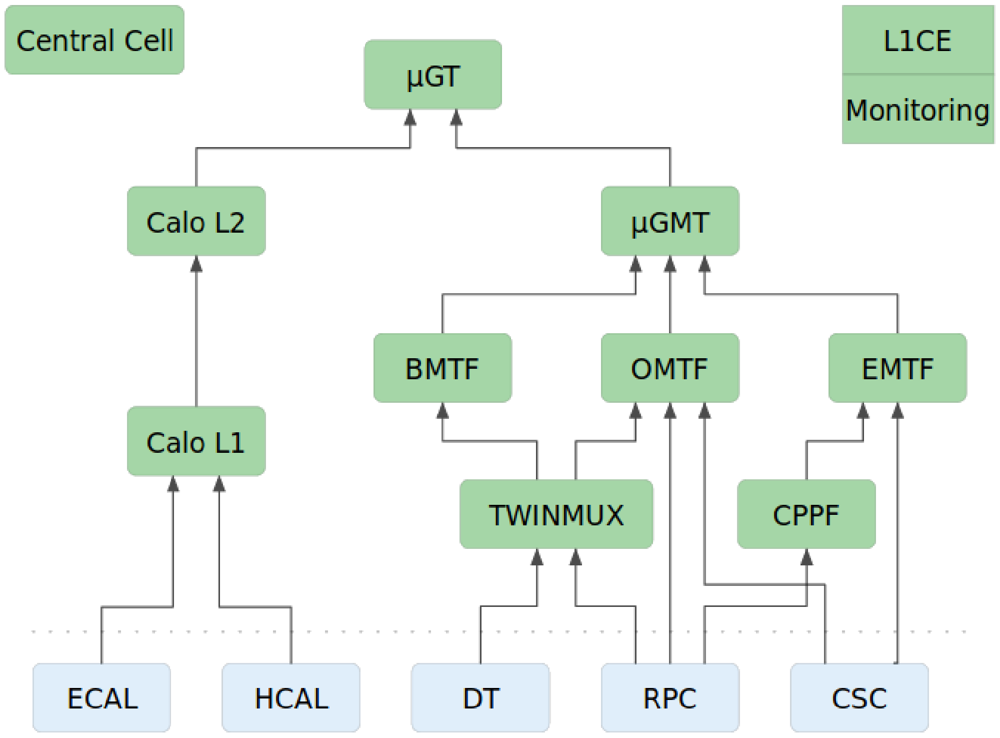


Figure 2.3: The Structure of The CMS L1-Trigger

The ”CALO-L1”, the ”TWINMUX” and the ”CPPF” are responsible to deliver the detector primitives to the Calorimeter Trigger ”CALO-L2” and the Muon Track Finders ”BMTF”, ”OMTF” and ”EMTF”. Each one of these systems reconstruct and deliver various local L1-particle candidates and global energy sums to the final stages of the L1T, the ”GMT” (Global Muon Trigger) and the ”GT” (Global Trigger).

CALO-L2 calculates sums of energy deposits or sums of missing energy and identifies physical objects like electrons, photos, jets and taus. After sorting them in terms of energy or transverse momentum it delivers 4 of each to the Global Trigger. In parallel to the Calorimeter Trigger, the Muon Track Finders (MUTF) identify muon signatures and calculate their momenta in each region they are responsible (Barrel-MUTF or BMTF, Endcap-MUTF or EMTF and Overlap-MUTF or OMTF - an overlapping region between the Barrel and the Endcap). After sorting them, all the three muon subsystems deliver their objects to the GMT. Last in the muon chain, the GMT rejects common muons that two subsystems had found and then sorts the muon candidates in terms of transverse momentum and quality. Finally, GMT delivers the eight highest muons to the GT.

The Global Trigger is the last layer of the L1 system. The main goal of the system is to produce the L1 decision and deliver it to the subsystems. The decision is calculated using the received input (candidates and sums) and also a list of the interesting combinations[1] of those inputs, the ”Physics Menu”. Also GT is capable of controlling the amount of positive L1 decisions by prescaling them on the fly. This functionality takes into account the needs and limitations of the next computing system in the chain (High Level Trigger) and the CMS storage elements.

[1] Interesting in terms of physics

Chapter 3

The Barrel Muon Track Finder

3.1 Introduction

The Barrel Muon Track Finder (BMTF) is a L1 Trigger subsystem whose task is to search and reconstruct muon candidates in the CMS Barrel Muon detector region (figure 3.1). BMTF receives data in the form of muon stubs from the TwinMux subsystem[1]. The Barrel Muon detector is defined by $|\eta| < 1.3$, however BMTF processes data coming from the region $|\eta| < 0.83$. The OMTF subsystem processes the rest barrel region up to $|\eta| = 1.3$.

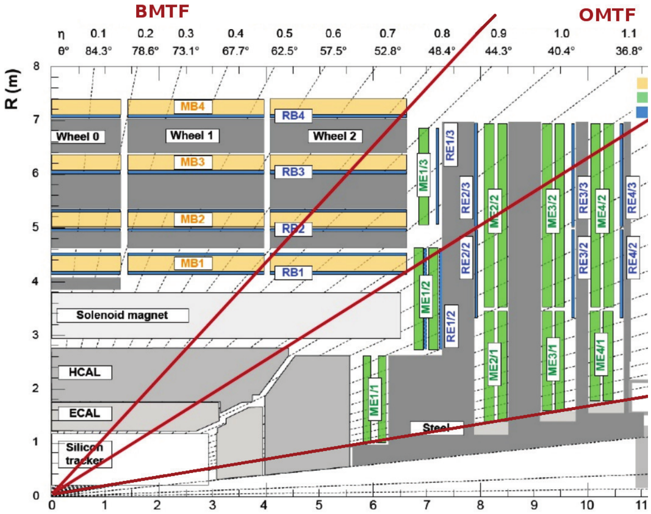


Figure 3.1: The three L1 Muon Track Finder Regions.

[1] The TwinMux input are DT stubs and RPC hit clusters which originate from muon detectors

Regional Muon Candidates are reconstructed using the stub data. For each candidate BMTF reconstructs its momentum (P_T), pseudorapidity (η) and azimuthial angle (ϕ), before transmits four of them to the Global Muon Trigger (GMT). The BMTF algorithm has undergone several upgrades since its conception. Recently an entirely new algorithm based on the Kalman Filter algorithm has been introduced and has been running in parallel with the main BMTF algorithm.

Both the algorithms that are described in the next sections are built in principle to process DT stubs. The Drift Tube Chambers cover the cylindrical area separated into 12 wedges of 30^o each. For each wedge there is a further separation into 5 sections (the η wheels). There are two groups of DT stations. For the ϕ-group, from inner to outer, exist 4 DT stations. For the η-group, 3 instead of 4 stations exist, while each of them is separated into 7 chambers.

3.2 The BMTF Phase-1 Algorithm

The BMTF algorithm is separated into 2 track finders, the Phi and Eta Track Finders. The implementation of the design in the FPGAs follows pipeline principles. This means that each step of the design is ready to process the next occurring event right after it has finished with the previous.

Figure 3.2 presents the main logic behind the Phase-1 algorithm. It is enumerated below, in short the algorithm is consisted by three steps. Extrapolation, track assembling and assignment of the variables (P_T, η, ϕ, etc). The Phase-1 algorithm has been described in the Technical Design Report for the Upgrade of the L1 Trigger [7].

1. The Phi Track Finder
   This logic block processes stubs coming from the ϕ-group of DT stations.

   (a) The Extrapolator
   For each combination of hits coming from 2 different stations, this block checks if the outer's station hit lies in between an acceptable window with respect to the inner hit. The edges of the windows' ranges, for all the combinations of stations, are stored in LUTs. In case of a positive decision this logical block sets a bit which indicates that this track element can be used to assemble a full track.

   (b) The Track Assembler
   This block connects compatible track elements to a full track and

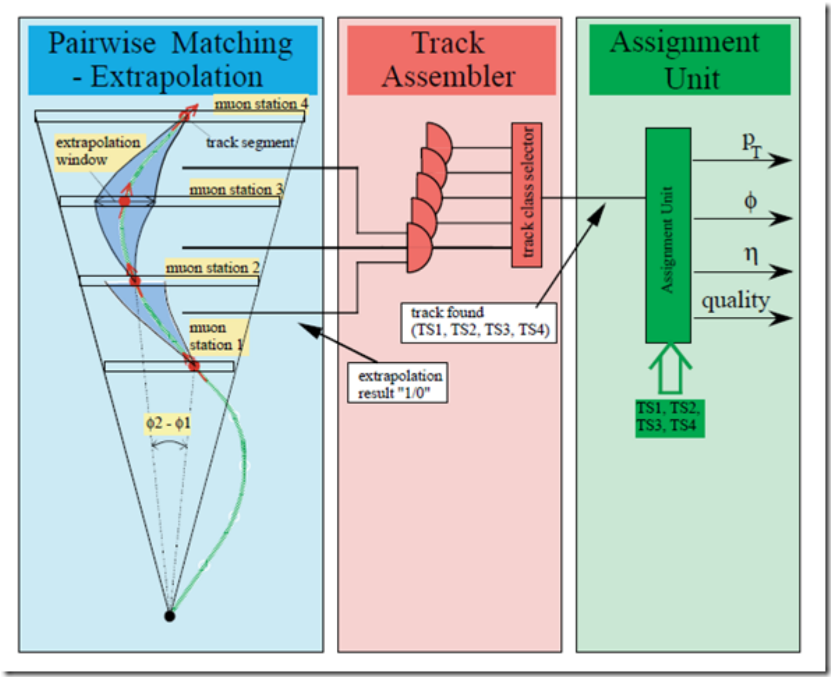


Figure 3.2: Presentation of the BMTF Phase-1 Algorithm.

calculates the Track Address. ”Track Address” (TA) is called one of the variables of the muon. It holds the information of the position of the track[2] to which the muon is related. At this point, information about the constructed phi tracks is propagated to the Eta Track Finder.

(c) Track’s Parameter Assignment
The generated Track Addresses are used as address to LUTs (pointers) which assign the variables (P_T, η, ϕ, etc) of the regional muons to each track.

2. The Eta Track Finder This logic block processes stabs coming from the eta stations.

   (a) Eta Pattern Network and Priority Grouping
   This logic block performs pattern recognition, on the input η hits, using LUTs. The patterns are grouped by hits’ quality, this way the matched η tracks are assigned with a priority value.

[2] Using the TA the next subsystem, μGMT performs a ghost cleaning of the muons produced by the neighboring track finders (BMTF-OMTF and OMTF-EMTF).

(b) The Priority Selector
According to the priority values of the tracks included in a quality group, the highest priority track is being selected for each group. In case of 2 tracks with same priority inside a group, the one with the lowest η value is selected.

(c) The Track Assignment
This block uses the highest priority tracks from each group and tries to match them with tracks received from the Phi Track Finder. If a match occurs a fine η is calculated, if not a rough η value is given that is associated with the Track Address and η-stub patterns and also the wheel information. In case of multiple matching, the rough value is preferred since there is no way to make the match unique.

3. The Wedge Sorter
Once the candidate muon tracks have been reconstructed, the Wedge sorter sorts the tracks in P_T and quality. The final order of the Regional Muon Candidates is at first related to their quality. For those of the same quality the higher P_T muons are placed higher in the order. Lastly, this block performs a primary ghost busting (removes duplicate tracks) before the three highest ranked muons are sent to the μGMT.

3.3 The BMTF Kalman Filter Algorithm

This is the most extensive upgrade of the BMTF system for 2018. The following algorithm illustrates a Kalman Filter optimized specifically to be implemented in the BMTF MP7 boards alongside the Phase-1 algorithm described in section 3.2. The algorithm has been optimized for the BMTF by the UCLA CMS team.

In general the Kalman filter (KF) is regarded as the optimal solution to a great class of tracking problems. For completeness of this document, some examples follow.

- Control and navigation of vehicles, for example autopilot systems and GPS.
- Signal processing, for example the speech enhancement by cancelling out several kinds of noise.
- Extraction of relations by large scale economic data samples.

- A variety of many other estimation driven applications (eg. seismology, weather forecast, radars, robotic motion, system's state estimation, etc)

In this section the principles of a Kalman Filter [8] are given and then they are used to illustrate the implementation in the MP7 boards. In the next paragraphs the four basic equations of the algorithm are presented and explained assuming that one faces a problem in which the signal that a system produces can be described by a state vector χ_k. For a derivation of the equations please look at the appendix. The general Kalman algorithm is displayed in figure 3.3

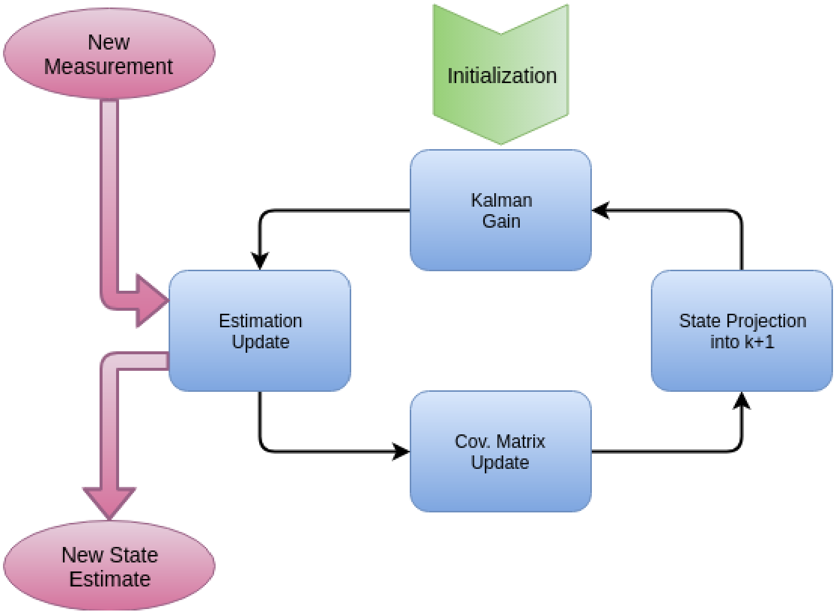


Figure 3.3: The Kalman Filter Loop.

The recursiveness of the algorithm lies in the iteration of four steps presented in the blue process boxes in figure 3.3 (for describing it here let's call it "the Kalman Filter loop"). The steps of the Kalman loop are:

- The Estimation Update.

$$\chi'_k = \chi_k + G_k(z_k - H\chi_k) \tag{3.1}$$

- The Covariance Matrix Update.

$$P'_k = (\mathbb{1} - G_k H)P_k \tag{3.2}$$

- The Projection of χ'_k and P'_k to the Next Input.

$$\begin{aligned} \chi_{k+1} &= \Phi\chi_k \\ P_{k+1} &= \Phi P_k \Phi^\top + Q \end{aligned} \tag{3.3}$$

- The Kalman Gain Calculation.

$$G_{k+1} = P_k H^\top (H P_k H^\top + R)^{-1} \tag{3.4}$$

Assuming that a prior estimate of the k^{th} signal (χ_k) and the related Kalman Gain (G_k) both exist (either by a previous iteration or via initialization), the first step is to use equation 3.1 to generate the updated estimate χ'_k taking into account also the newly inserted measurement. The k^{th} measurement is described by the state vector z_k, which is also a representation of the signal in the most sensible space[3] (becomes clearer in the next paragraphs via the application on the BMTF). The matrix H is a mapping between those two spaces, $z_k = H\chi_k$. After the estimation update, the update of the covariance matrix P_k takes place using equation 3.2. P_k is the covariance matrix of the signal expressed as the estimated value of the difference between the updated estimation and the prior prediction.

$$P_k = E[(\chi'_k - \chi_k)(\chi'_k - \chi_k)^\top] \tag{3.5}$$

Once both updates have been completed, next one needs to project the updated estimations to the next signal input. This is a transformation of the vector χ'_k and the matrix P'_k inside the signal space using the matrix Φ. The required two equations are displayed in 3.3. Φ is a transformation matrix which describes the geometry and specific features of each problem. The last step of the Kalman loop is the Kalman Gain recalculation for the next signal, equation 3.4. The quantities Q and R in this context are covariance coefficients due to signal's noise. In signal processing, signals frequently suffer from white noise which can be described by an additive term in the signal's description[4].

For the BMTF the produced signals are the DT/RPC stubs received by the TWINMUX. An input stub ”k” can be described by the angle ϕ_k and the bending angle ϕ_{bk} together in a state vector z_k. The output of BMTF is the measured P_T (among other variables), related to the reconstructed muon's

[3] For example, in BMTF a signal is a hit which is described by two variables ϕ and ϕ_b, so z_k consists from these two variables, while χ_k includes also the curvature $\tilde{k}$ of the muon's trajectory passing by the stub ”k”.

[4] $y_k = \alpha_k \chi_k + n_k$, χ is signal's waveform, α is signal's amplitude and n is the noise.

trajectory. The curvature $\tilde{k}$ of the trajectory is a function of $1/P_T$. The abstract space χ_k includes also the trajectory's curvature at a stub's point, since using it one can estimate and update the P_T measurement by estimating the state vector of the next $(k+1)$ signal in a Kalman Filter fashion. Equation 3.6 presents the two signal representations and their mapping.

$$\chi_k = \begin{pmatrix} \tilde{k} \\ \phi \\ \phi_b \end{pmatrix}_k , \quad z_k = \begin{pmatrix} \phi \\ \phi_b \end{pmatrix}_k , \quad H = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.6}$$

An approximation of the described algorithm has been implemented in the BMTF boards. The four steps can be reduced into two if one calculates in advance the required values of the Kalman Gain (G_k) for a specific scenario. The reason lays to the fact that the matrix P_k is required only for the Kalman gain recalculations, hence the update step of the covariance matrix can be omitted if the next gain exists stored. The necessary G_k values can be stored in BRAMs of the FPGA and can be fetched according to the update in hand.

1. Track Propagation
   In this step the state vector χ_k, describing a prior stub, is being propagated to the next stub $k+1$ from the outer station to the inner. This propagation is the step described by equation 3.3, and the required transformation matrix[5] Φ is presented in equation 3.7.

$$\begin{pmatrix} \tilde{k} \\ \phi \\ \phi_b \end{pmatrix}_{k+1} = \begin{pmatrix} 1 & 0 & 0 \\ \alpha & 1 & \beta \\ \gamma & 0 & \delta \end{pmatrix} \begin{pmatrix} \tilde{k} \\ \phi \\ \phi_b \end{pmatrix}_k \tag{3.7}$$

   The required multiplications for this step have been implemented using the DSP cores[6] of the Virtex 7 which is featured on the MP7 boards.

2. State Vector Update
   In this step the appropriate stored Kalman gain is used to update the stub estimation performed by Step-1. Kalman gain values have been pre-calculated as a function of the curvature $\tilde{k}$ and a 4-bit hit pattern. The pattern formulates the stations with stubs, one bit for each station (S4-S3-S2-S1). Different combinations of the pattern are

[5] This is a transition matrix which takes the position of the prior stub laying on the parabolic trajectory and projects it to the next stub via the parabolic equation. The calculation has been done by the UCLA team.

[6] Specialized cores embed in the FPGAs for digital manipulation of signals. They are capable of handling multiplications.

related to different input stub scenarios. For example "1011" means stubs at the stations 1-2-4, while "0101" means stubs at the stations 1-3. The following equation, similar to 3.1, updates the propagated estimation taking into account the new stub z_k.

$$\chi'_k = \chi_k + G_k(\tilde{k}, \text{4bit-pattern}) \times [z_k - H\chi_k] \tag{3.8}$$

The required multiplications are being done by DSP cores. The Kalman Gains are stored in Block-RAMs embed in the Virtex 7s. This step is skipped if a stub doesn't exist on the station in hand , so one would have two propagate steps in sequence worsening the χ^2 merit function (see below) with respect to the better track fits.

3. Propagation to the Vertex and Update
Once the iteration of the Kalman loop over the four stations has been finished, the algorithm propagates the track to the vertex by constraining the trajectory to pass by the center of CMS. However since the measurement and the uncertainties of the variables exist before the vertex propagation, one can store and use them as displaced measurements for the reconstructed muons. Hence the Kalman algorithm provides two P_T measurements for each muon, a displaced and a vertex constrained.

The Kalman loop approximated by steps 1 and 2 is implemented in each BMTF board 22 times. All the possible track scenarios (described by the 4-bit pattern) are being processed simultaneously in each processor. The number 22 originates from the combinations of the tracks with at least two stubs in the four stations(11), multiplied by 2 because each station can deliver up to two stubs.

$$\binom{4}{2} + \binom{4}{3} + 1 = 6 + 4 + 1 = 11$$

Kalman Filter's objective is to provide the best estimate of the signal in a "mean squared error" sense. It is possible to define a "χ-squared" merit function[7] in order to rank the result of the filter's output. The BMTF Kalman algorithm uses a χ^2-cut in order to select the best tracks out of those that overlap.

One last comment is appropriate regarding the Q and R noise matrices. In the BMTF variation of the Kalman algorithm these matrices can be used to formulate the multiple scattering effects. The matrices are compatible with errors which can be described using Gaussian distributions. This model is convenient to be used for the measurement uncertainties which

[7] Look at the appendix, $\chi^2 = (\chi_k - \chi'_k)^\top P_k^{-1}(\chi_k - \chi'_k) + [z_k - f(\chi'_k)]^\top R^{-1}[z_k - f(\chi'_k)]$.

occur mainly while a muon propagates between the stations, resulting into worse resolution of the muon variables. The covariance coefficients related to multiple scattering effects have been pre-calculated using Monte-Carlo simulation (by UCLA) and the information is being held by the pre-calculated Kalman Gains.

3.4 BMTF Implementation at P5

The BMTF subsystem is implemented on 12 MP7 and 2 AMC13 boards. The boards are organized in two crates. One AMC13 and six MP7s are placed in one crate. For each AMC13 there is one DAQ FED which is used to control the system centrally. For the BMTF the two corresponding FED-id numbers are 1376 and 1377. The MP7 boards implement the algorithms, while AMC13s handle the readout and procedure and the crate management. Each MP7 board receives from the TWINMUX η and ϕ stubs originated by three successive wedges[8]. The input links for one MP7 are 30, two links per wheel and a total of 10 per wedge[9], while the output links are 2, one per algorithm.

The main triggering algorithm (Phase-1, which had been used during 2016 and 2017) and the Kalman algorithm (which has been fully commissioned for Run 3) run in parallel both in the same FPGAs. The new algorithm is not triggering, it is running parasitically since Summer 2018. It is being fed with the same inputs as the old algorithm, and it's being read out.

The BMTF team has also provided supporting software, which is installed and running at the CMS Control Room machines (at the LHC-P5). At P5 there are 2 monitoring software solutions, one of them is monitoring the hardware directly (Online Software), and the other is monitoring the data as they are collected (Offline Software, the DQM). The Software is described in detail in the next section and also there is a dedicated chapter following.

3.5 The BMTF Online Software

This is the control software of the subsystem's (BMTF) FPGAs. It's capable to configure, control and (while it is running) monitor the BMTF system. The BMTF Online Software is an implementation of a "Swatch Cell" (Soft-

[8] A wedge is a DT chamber cluster which covers 30^o in ϕ and entire η.

[9] Each board processes three wedges because muons may cross from one central wheel to the next due to the bending force applied by the magnet.

Ware for Automating the conTrol of Common Hardware[10]). The Swatch Cell is a package that is designed to communicate via the IPv4 protocol with the boards that are being used for the purpose of the L1 Trigger project. This is being done by reading and writing to the boards' registers.

The BMTF Swatch Cell is capable of configuring the system using information written in xml files, the keys. Briefly, the keys that each Trigger subsystem uses are four, the ”hardware” key, the ”infrastructure” key, the ”algorithm” key and the ”run settings” key. These four xml files hold the required parameters that the Swatch Cell uses for the system's description and configuration (a longer description of the keys can be found in the appendix). Using the Cell one can control entirely the transitions of the system's finite state machine. Also, via its register-based monitoring, the BMTF Cell is capable of issuing ”Warning” and ”Error” messages to the control room of the CMS experiment at P5.

The finite state machine of the BMTF is displayed in figure 3.4. The state machine consists of 6 states (”Halted” excluded[11]) and 11 transition steps. Operation begins with the ”Engaged” state, this is the initial state when the system is handled by the Online Software once a ”hardware” key is loaded. Next comes the ”Synchronized” state, in which the boards have been rebooted, given a board id and loaded with the desired firmware from the sd card. The ”Configured” state follows in which the inputs are being zeroed, registers that must contain algorithm and infrastructure parameters are written with the correct information, the latency of the algorithms is defined, and the read-out and zero-suppression menus are loaded. After the Configured state the ”Aligned” takes place, in which the board transceivers are getting configured and aligned. When in the Aligned state the system is capable of either going to the ”Running” or the ”Configured” state. Last state is the ”Paused” state. This is used if the system is already running and a change needs to be done but the run needs to continue.

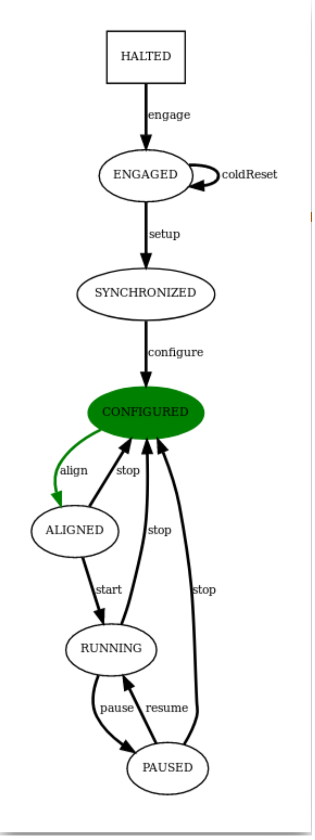
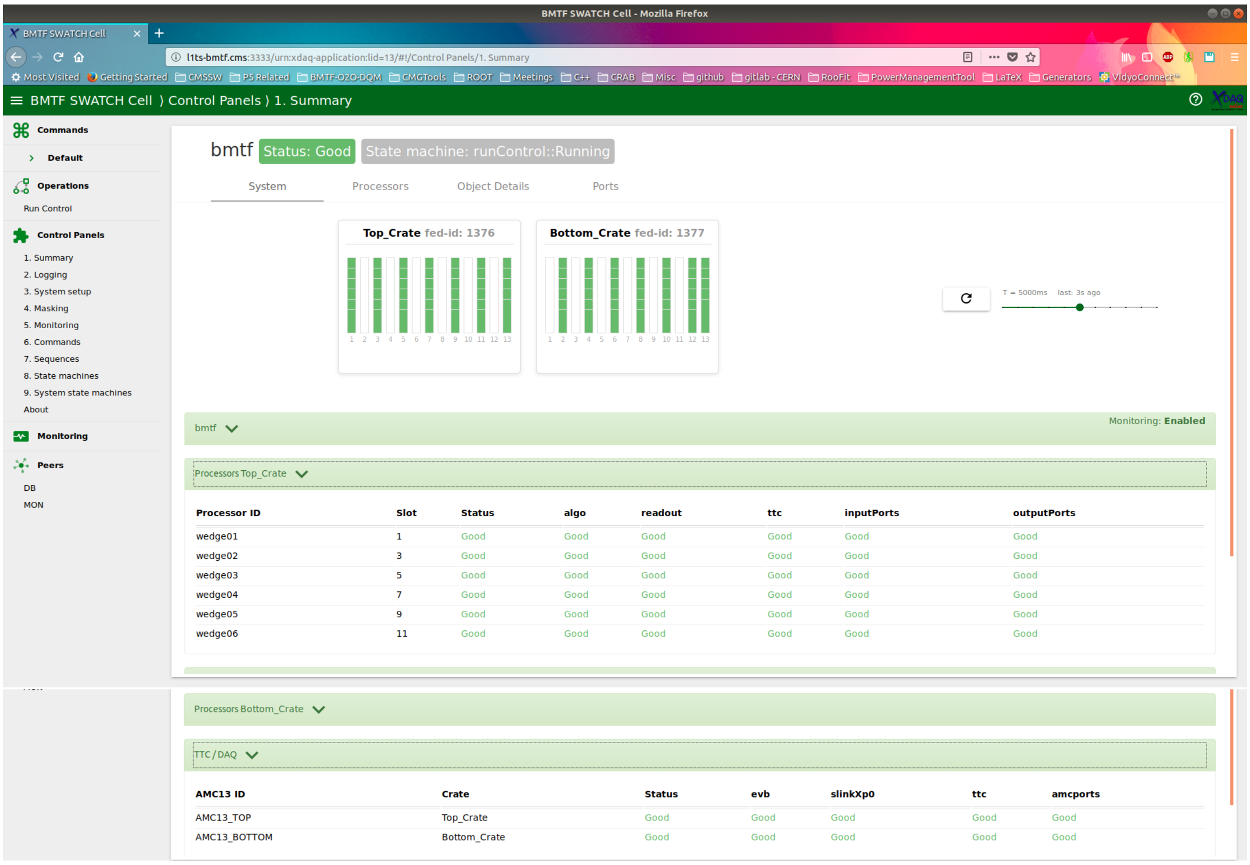


Figure 3.4: The BMTF State Machine.

[10] The website given here [9], is the official page of the developer team including the release notes and also a developer's guide.

[11] Halted is a stopped state in which the system enters after initialization of a new run.

Except the runtime functionality and the handling of the system, the online software also offers a panel for running commands to perform several low and high level operations (like scanning SD cards, uploading firmware, etc.). Five screenshots are described next, these are the "Summary Panel", "the "Run Control Panel", the "Monitoring Panel", the "Commands Panel" and the "System Setup Panel". These five panels seem to be the most useful ones that someone needs when operates the BMTF system.



bmtf Status: Good State machine: runControl::Running

Top_Crate fed-id: 1376

Bottom_Crate fed-id: 1377

bmtf

Monitoring: Enabled

Processors Top_Crate

Processor ID	Slot	Status	algo	readout	ttc	inputPorts	outputPorts
wedge01	1	Good	Good	Good	Good	Good	Good
wedge02	3	Good	Good	Good	Good	Good	Good
wedge03	5	Good	Good	Good	Good	Good	Good
wedge04	7	Good	Good	Good	Good	Good	Good
wedge05	9	Good	Good	Good	Good	Good	Good
wedge06	11	Good	Good	Good	Good	Good	Good

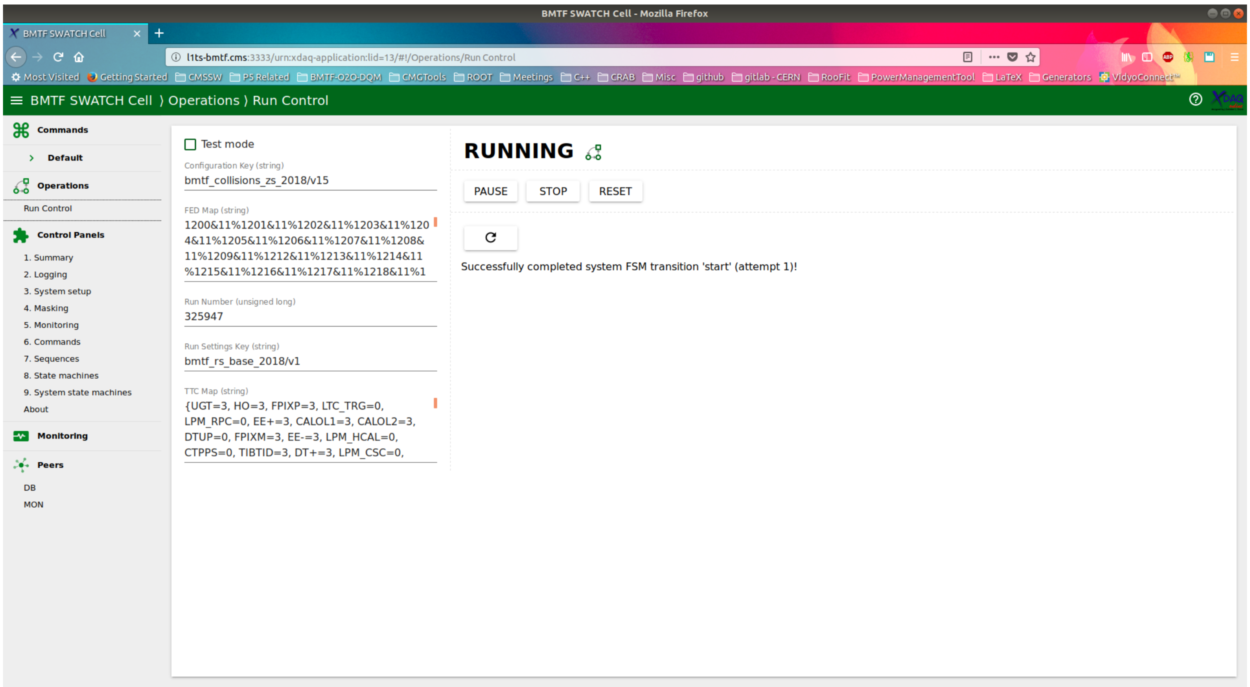
Processors Bottom_Crate

TTC / DAQ

AMC13 ID	Crate	Status	evb	slinkXp0	ttc	amcports
AMC13_TOP	Top_Crate	Good	Good	Good	Good	Good
AMC13_BOTTOM	Bottom_Crate	Good	Good	Good	Good	Good

Figure 3.5: The Swatch Summary Panel.

Figure 3.5 displays the panel which has a summary of the current (when running) condition of the system. There is a visual representation of the two BMTF crates, with their names and also their FED numbers. The FED numbers can be used directly by the DAQ Shifter at P5 to include or exclude the BMTF subsystem from the runs. As is already described in the previous section there are seven boards in each crate (6 MP7s and 1 AMC13). Their running condition is being summarized in these lines, one for each processor ("wedge") and two more for the AMC13s. The first column is the processor name, while the next to the right is its place in the crate. The rest columns, from left to right, display the general status of several registers of the board (which are being monitored) clustered together in six categories for the MP7 and five for the AMC13.



□ Test mode

Configuration Key (string)
bmtf_collisions_zs_2018/v15

FED Map (string)
1200&11%1201&11%1202&11%1203&11%1204&11%1205&11%1206&11%1207&11%1208&11%1209&11%1212&11%1213&11%1214&11%1215&11%1216&11%1217&11%1218&11%1

Run Number (unsigned long)
325947

Run Settings Key (string)
bmtf_rs_base_2018/v1

TTC Map (string)
{UGT=3, HO=3, FPIXP=3, LTC_TRG=0, LPM_RPC=0, EE+=3, CALOL1=3, CALOL2=3, DTUP=0, FPIXM=3, EE-=3, LPM_HCAL=0, CTPPS=0, TIBTID=3, DT+=3, LPM_CSC=0,

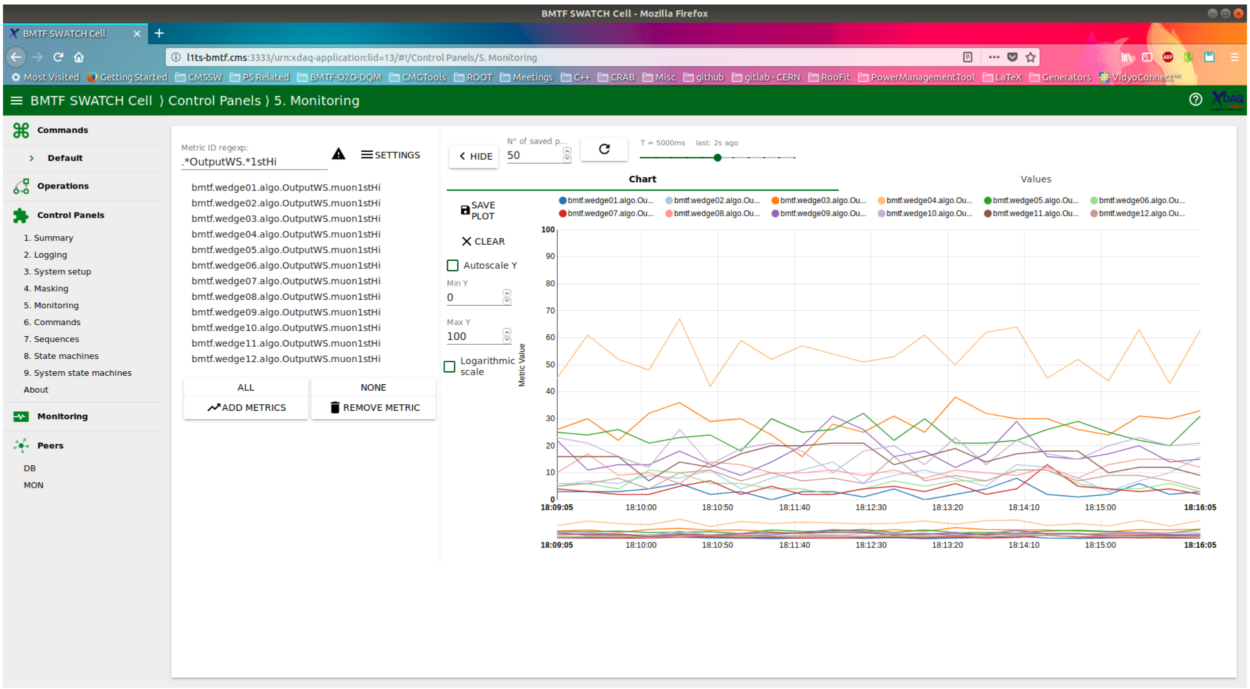
Successfully completed system FSM transition 'start' (attempt 1)!

Figure 3.6: The Swatch Run Control Panel.

Figure 3.6 displays the panel which displays the general handler of all the loaded boards simultaneously. On top, with big bold font, is reported the current state of the state machines. Below, the appearing buttons, are given the possible transitions. By pressing one of them, a given transition is being executed for all the engaged boards. On the left there are several parameters which can be used to engage an instance of the system using a Configuration[12] and a Run Settings key, directly from the L1 database (l1ce.cms). The rest required information is the FED mapping with other subsystems (TWINMUX and uGMT), the run number to be used and .

Figure 3.7 displays the panel in which one can monitor in a graph specific selected metrics (registers). Using this interface one can select all the desired metrics required for troubleshooting and plot their values as they are varying

[12] hardware, infrastructure and algorithm keys in one



Metric ID regexp: .*OutputWS.*1stHi

bmtf.wedge01.algo.OutputWS.muon1stHi
bmtf.wedge02.algo.OutputWS.muon1stHi
bmtf.wedge03.algo.OutputWS.muon1stHi
bmtf.wedge04.algo.OutputWS.muon1stHi
bmtf.wedge05.algo.OutputWS.muon1stHi
bmtf.wedge06.algo.OutputWS.muon1stHi
bmtf.wedge07.algo.OutputWS.muon1stHi
bmtf.wedge08.algo.OutputWS.muon1stHi
bmtf.wedge09.algo.OutputWS.muon1stHi
bmtf.wedge10.algo.OutputWS.muon1stHi
bmtf.wedge11.algo.OutputWS.muon1stHi
bmtf.wedge12.algo.OutputWS.muon1stHi

N° of saved p... 50

T = 5000ms last: 2s ago

□ Autoscale Y

Min Y 0

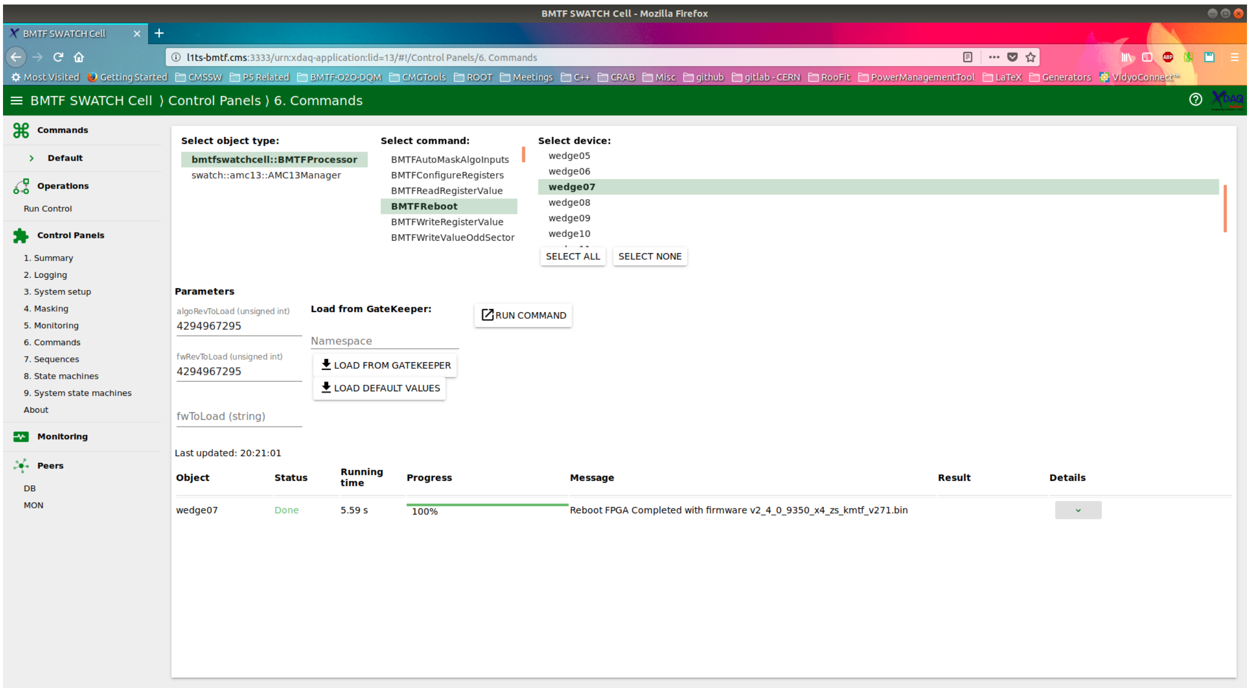
Max Y 100

□ Logarithmic scale

Figure 3.7: The Swatch Monitoring Panel.

in the graph area on the right. When plotting values one can configure the graph using the parameters on its right, max/min Y values and logarithmic scale of the Y axis choices are given. Also, the choice to save a copy of the plot is given, via the ”Save Plot” button on the left top of the graph. Last functionality of this panel to refer to, is the capability to display the values of the selected metrics organized in a list, by selecting the tab ”values” instead of ”chart” on the top of the graph.

Figure 3.8 presents the Commands panel. The target of this panel is to deliver the user a clean interface that issues commands to the boards. Beginning from the top left, one needs to choose the type of the device to which he (or she) needs to run a command. Then select the command to run and the target device from the two lists of the commands and boards related with the chosen device type. Once the command has been chosen, the bottom area of the panel will be filled (like in the picture) with the dialog that is used to load the parameters that are required for the command to run. There are three ways to load the parameters. Firstly, using the slots on the left to insert custom values. Secondly, using the button ”Load from Gatekeeper” to load the parameters from the loaded Infrastructure key. And thirdly, using the button ”Load Default Values” to use the command defaults if any. Finally, by pressing the ”Run Command” button the command is executed to the



Select object type:
bmtfswatchcell::BMTFProcessor
swatch::amc13::AMC13Manager

Select command:
BMTFAutoMaskAlgoInputs
BMTFConfigureRegisters
BMTFReadRegisterValue
BMTFReboot
BMTFWriteRegisterValue
BMTFWriteValueOddSector

Select device:
wedge05
wedge06
wedge07
wedge08
wedge09
wedge10

Parameters

algoRevToLoad (unsigned int)
4294967295

fwRevToLoad (unsigned int)
4294967295

fwToLoad (string)

Load from GateKeeper:
Namespace

Last updated: 20:21:01

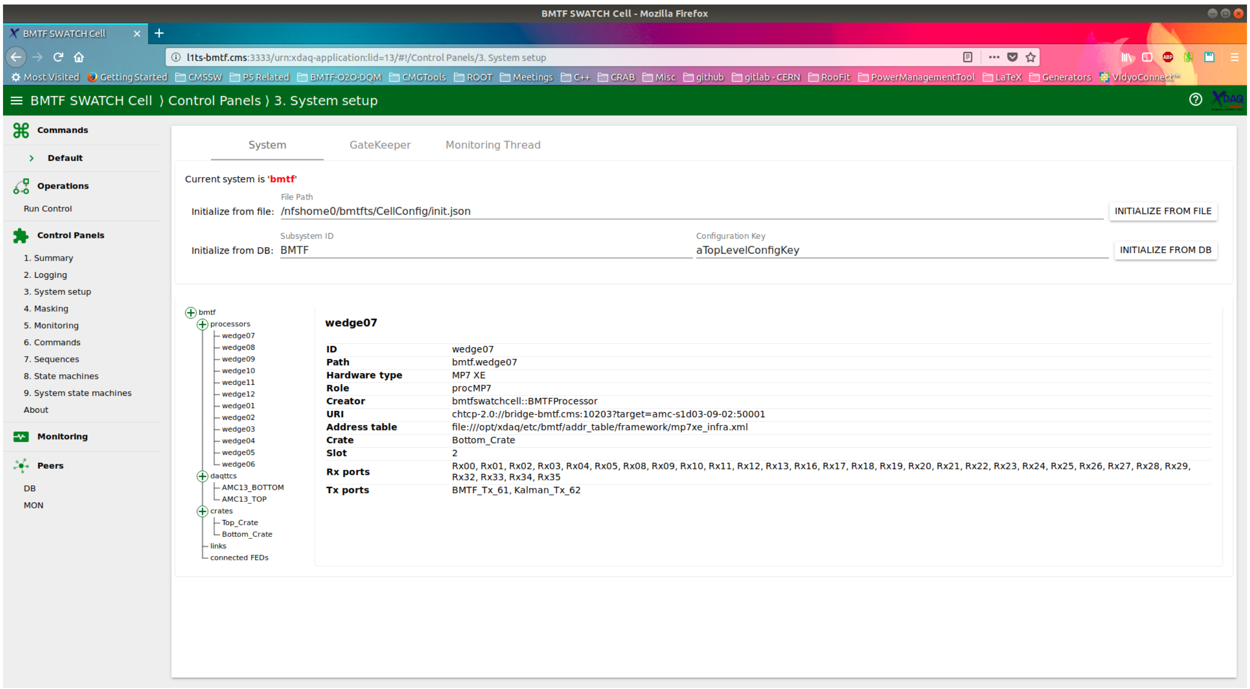
Object	Status	Running time	Progress	Message	Result	Details
wedge07	Done	5.59 s	100%	Reboot FPGA Completed with firmware v2_4_0_9350_x4_zs_kmtf_v271.bin		⌄

Figure 3.8: The Swatch Commands Panel.

selected board (or boards) and the progress is reported at the bottom of the panel.

The last Swatch panel is presented in figure 3.9. This panel offers the capability to load a system instance to the Swatch Cell manually. For loading a system instance one needs to point within the online software environment two keys, either locally or using the L1 database. The loaded system will be displayed (like in the picture above) using the tree-fashion graph on the bottom area of the panel. Also, by clicking the leaves of the tree on the left, a summary of the selected object will be appeared on the right.

The procedure to load a new system instance includes two steps. Firstly, in the above displayed ”System” tab, one needs to use a local hardware key or a database stored configuration key in order to load the description of the system. Once this has been done the tree representation of the system will be displayed. The keys are being loaded by selecting the appropriate files using the dialogues on the top and pressing the related ”Initialize ...” button. This is enough in case one needs only to run manually commands (one-by-one) using the Commands Panel presented in the figure 3.8. However, if the target is to run through the entire BMTF state machine one needs the parameters to be loaded for all the commands included in the FSM transitions. This is possible by loading the rest of the keys, using the ”Gatekeeper” tab. In the



Current system is 'bmtf'

Initialize from file: File Path /nfshome0/bmtfts/CellConfig/init.json

Initialize from DB: Subsystem ID BMTF Configuration Key aTopLevelConfigKey

wedge07

ID	wedge07
Path	bmtf.wedge07
Hardware type	MP7 XE
Role	procMP7
Creator	bmtfswatchcell::BMTFProcessor
URI	chtcp-2.0://bridge-bmtf.cms:10203?target=amc-s1d03-09-02:50001
Address table	file:///opt/xdaq/etc/bmtf/addr_table/framework/mp7xe_infra.xml
Crate	Bottom_Crate
Slot	2
Rx ports	Rx00, Rx01, Rx02, Rx03, Rx04, Rx05, Rx08, Rx09, Rx10, Rx11, Rx12, Rx13, Rx16, Rx17, Rx18, Rx19, Rx20, Rx21, Rx22, Rx23, Rx24, Rx25, Rx26, Rx27, Rx28, Rx29, Rx32, Rx33, Rx34, Rx35
Tx ports	BMTF_Tx_61, Kalman_Tx_62

Figure 3.9: The Swatch System Setup Panel.

Gatekeeper tab the dialog fashion is the same, above there is the dialog to point local files and below the dialog to point database key records.

During the last year there have been some minor developments for the BMTF Swatch Cell. The changes are enumerated in the list bellow.

- Warning and Error runtime thresholds changed.
- Monitorables' names have been slimmed down regarding their string length.
- Reboot command extended to load the requested firmware.
- Configuration sequence upgraded to support two algorithms in parallel.

The first two upgrades performed during the winter shutdown between runs 2017 and 2018. The thresholds are hardcoded conditions which are being tested against registers' values in order to raise runtime warning and error messages. It was necessary to be changed because of modifications of the L1 system's configuration which raised the rates of some L1 algorithms. The names of the monitorables are used also to store the configuration of the systems in the online database. It had been requested to slim down their length in order to be compatible with some new database rules (for example names' length up to 30 characters).

The modification of the configuration sequence has been done on Summer 2018. Because of the different latency of the Phase-1 and the Kalman algorithms, it is necessary to configure them with different parameters each. The parameters in question are not related to the algorithm setup, instead they are related to the MP7 functionality regarding the readout and synchronization procedures. From software perspective a group of commands must be run twice, once for each algorithm, and also each group of commands must be declared that belongs to one algorithm or the other.

Last upgrade is the reboot command upgrade, on Fall 2017. The command is used in the configuration sequence. This development has already contributed at the most during the Kalman Algorithm's testing. Before this modification the naming scheme of the firmware files was specific, and the BMTF Cell was configuring the system using the latest saved file. Now, the Swatch Cell is capable of searching for a requested (by the key) file and use this during the booting process of the boards. Having this flexibility the testing procedure of any new firmware is being simplified. A new testing key is enough, and the testing firmware can remain stored in the SD cards waiting for offline analysis of the test run that has been collected.

```
<infra id=”bmtf”>
  <context id=”procMP7”>
    <param cmd=”BMTFReboot” id=”fwToLoad” type=”string”>
  v2_4_0_9503_x4_zs_kmtf_v271.bin</param>
    <param cmd=”BMTFReboot” id=”algoRevToLoad” type=”uint”>0
  x95030160</param>
    <param cmd=”BMTFReboot” id=”fwRevToLoad” type=”uint”>0
  x12020400</param>
```

Except the offered ease, this modification has been offered also extra cross-checking regarding the keys. In the firmware, there is hardcoded information about its version. So, aside the firmware searching functionality which added to the Cell, a procedure has been added before the software reboots the boards to check for possible human mistakes. The BMTF Cell checks if the hardcoded algorithm version and the hardcoded MP7-infrastructure version match to what is given by the key. And along with the searching for the requested firmware in the SD cards, this gives us a new three-point cross-check procedure.

Chapter 4

Offline Software Support for BMTF

4.1 Introduction

The CMS collaboration has developed over the years a powerful framework for handling and analysing the data collected by the experiment, the CMS Software (CMSSW). The structure of the software is organised into packages and each of them in several modules, while the packages support plugins as well. The modules are written in C++, and each of them operates by requiring at least one object as input (the terminology calls this a ”consumable”). The framework uses the concept of the Events (edm::Event class). This is a C++ container which holds information about the collision, and is responsible of propagating this information to all the modules.

There are six different classes of modules that are defined in the context of the CMSSW. The concept of the software is illustrated by the drawing in figure 4.1. The schematic consists of the Source and the Output Pools, the EDProducer, the EDAnalyzer, the EDFilter and the EDLooper. All the six of them accomplish different needs of the analysis procedure. The purpose of the two pools is the data input and output handling. The Source Pool initializes and the Output Pool destructs (terminates) the Event Container. These reading and storing procedures can be accomplished using either the local storage of the machine on which the CMSSW release is installed, either the online CMS Data Aggregation System (CMSDAS) over the network via read and publication queries.

The rest four modules are responsible for the data processing. Modules are organized into a ”path”, a sequence of modules which defines in what order they will loop over the entire collection of the events. An EDProducer is a module that reads (consumes) objects stored in the event and produce new objects which will be used after from other modules. An EDAnalyzer is a module whose purpose is purely the analysis, and is not capable of storing anything to the event container. It produces plots or variables that will be stored in an external ROOT file. An EDFilter is a module responsible for making the decision if the event in hand needs to be omitted, and so the execution of the next modules in the path can be skipped. EDFilters return a boolean to be handled by the framework and do not store anything to the event container. Last class of modules is the EDLooper. This module operates very similar to the EDProducer but it is able to loop over the events more than once. All these four classes of modules are capable of storing information in a ROOT file as well.

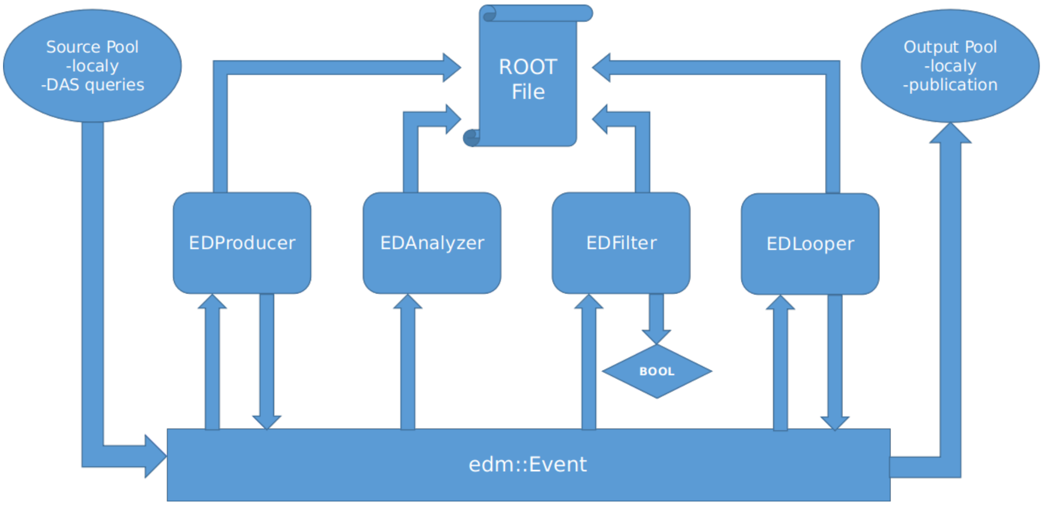


Figure 4.1: The CMSSW Concept.

The entire collection of source code is hosted in an online repository at github.com[1]. More information about the CMSSW can be found here [10]. The Barrel Muon Track Finder (BMTF) has its own modules and several plugins in order to handle the data acquisition procedure and also the data quality and performance control. Split into three groups these modules are: (a) 2 Emulators, (b) 2 Unpacker and 2 Packer plugins and (c) several more plugins for the O2O and DQM packages (the terminology used here is explained throughout the chapter). This chapter is dedicated in describing the BMTF software and the latest developments.

[1] The repository can be found at https://github.com/cms-sw/cmssw.

4.2 Data Acquisition

During the data taking procedure, several CMSSW modules run in parallel to assure and monitor the data acquisition process. One of the most vital modules since its construction is the Data Quality Monitoring (DQM) module. It has contributed vitally to all procedures and tests because it presents, as the experiment proceeds, readout plots for the subsystems and comparison between firmware and emulator results. Proper functioning of the DQM requires input from several other modules, some of which are presented below.

4.2.1 The Unpacker Plugins

The BMTF readout data record is in binary format (see figure 4.2). In CMSSW this binary information is included in an object stored in the Event container, the "FEDRawDataCollection". The Unpacker[2] is responsible to translate this binary information to C++ objects and making the information capable to be handled by other modules (like the Emulators). Two plugins exist for the BMTF Unpacker, one of them handles the input data and the other handles the output data.

The unpacking procedure is handled by a L1-general "supermodule" (figure 4.3). This supermodule is responsible to extract the required information from the FEDRawDataCollection and run correctly those two BMTF plugins. Each plugin defines abstractly the behavior of the corresponding "BMTFUnpackerInputs" or "BMTFUnpackerOutput" C++ class. Objects related to these classes will be handled by the supermodule in order to unpack the raw data coming from a specific MP7. For each MP7 two unpackers are required, one for the 10 own[3] input links and one else for the 2 output links.

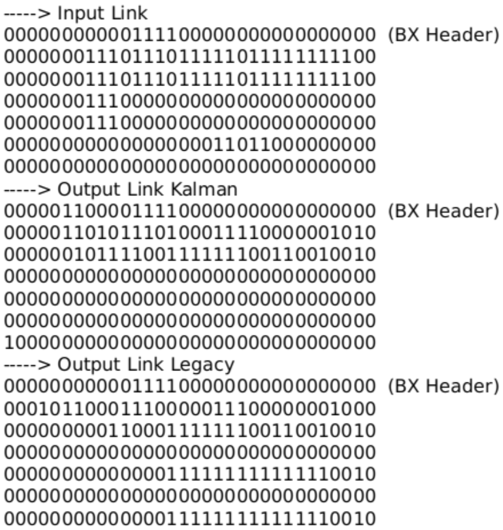
After the declaration of the necessary unpackers, the unpacking supermodule, loops over the related FEDs for the L1 subsystem (BMTF owns 2 FEDs, 1376 and 1377, each including information of six processors). For each FED there is a nested loop over the related MP7s and the corresponding unpackers are being executed in order to unpack their input and output links. Once this procedure is finished, the supermodule delivers to the event container three new objects, the "L1MuDTChambPhContainer", the "L1MuDTChambThContainer" and the "RegionalMuonCandBxCollection". The first two hold the ϕ and η hits, while the third is a "BXVector" of L1

[2] The related package for the L1 subsystems is the "EventFilter/L1TRawToDigi" and the BMTF plugins can be found under path "plugins/implementations_stage2".

[3] "Own links" are called those related to the processor's specific inputs, not the left or the right processor's input, because each board reads out all the information that processes.

muon candidates[4] holding their hardware variables.

The FEDRawDataCollection includes the BMTF input and output payloads organized in an object called ”Block”. A FED Block is shown in figure 4.2. Visually one could think of a block as a sequence of 32-bit words coming from a specific link and processor. Another similar object is the ”BxBlock”, which is also being used by the CMSSW and is a sequence of 32-bit words related to a specific bunch-crossing and processor (this is actually what is displayed in figure 4.2). The unpacker objects are executed by feeding them with link-Blocks and other useful information, like the processor ID and the firmware version.



```
-----> Input Link
00000000000111100000000000000000  (BX Header)
00000001110111011111011111111100
00000001110111011111011111111100
00000001110000000000000000000000
00000001110000000000000000000000
00000000000000000011011000000000
00000000000000000000000000000000
-----> Output Link Kalman
00000110000111100000000000000000  (BX Header)
00000110101110100011110000001010
00000010111100111111100110010010
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
10000000000000000000000000000000
-----> Output Link Legacy
00000000000111100000000000000000  (BX Header)
00010110001110000011100000001000
00000000011000111111100110010010
00000000000000000000000000000000
00000000000000111111111111110010
00000000000000000000000000000000
00000000000000111111111111110010
```

Figure 4.2: The BMTF readout record.

Figures 4.4 and 4.5 present the algorithms which used by the two BMTF unpacking plugins. Both plugins, have been through serious development integrating the following three upgrades. Under the changes the same scheme of the code has been preserved.

- Support for the Zero Suppression capability of the BMTF subsystem.
- Extension of the logic to cover also the Kalman readout.
- Support for the dynamical switching capability of the triggering algorithm.

The flowchart 4.4 illustrates the inputs' plugin. The algorithm begins by checking if the received block originates from an ”own” link, and if not the block is skipped. For the blocks coming from own links the ϕ and η containers for the products are being created and registered. Next comes the ”Zero Suppression Handler” (ZS). This piece of code is responsible to understand if the ZS[5] is activated and then to extract the BxBlocks from the Block object

[4] In other words RegionalMuonCandBxCollection is a std::vector of RegionalMuonCand.

[5] The Zero Suppression (ZS) operates in a BX-per-BX mode. It suppresses ONLY BxBlocks in case they don't hold hits or muons.

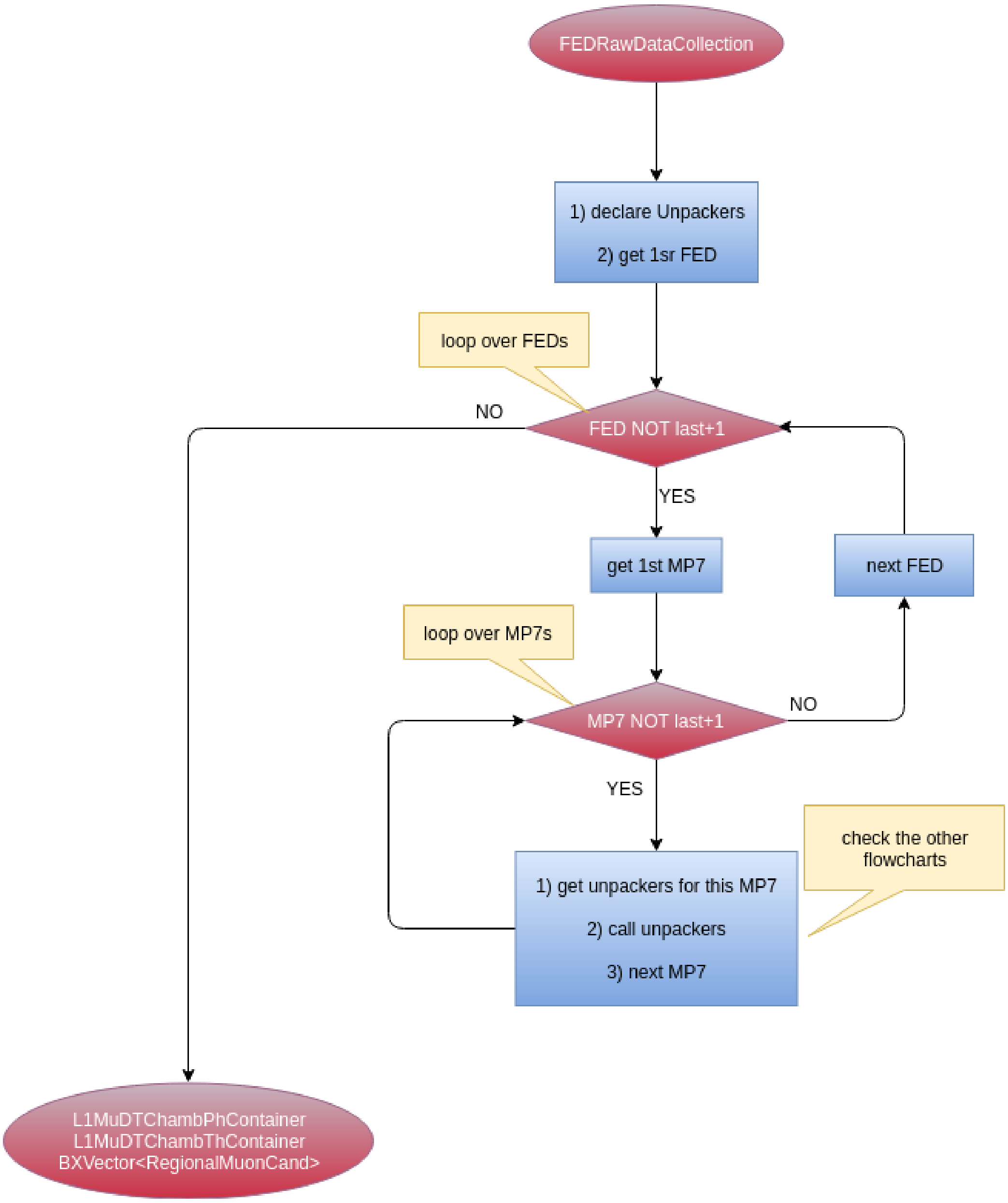


Figure 4.3: The L1-Trigger Unpacking Supermodule.

Figure 4.4: The Inputs' BMTF Unpacker Plugin.

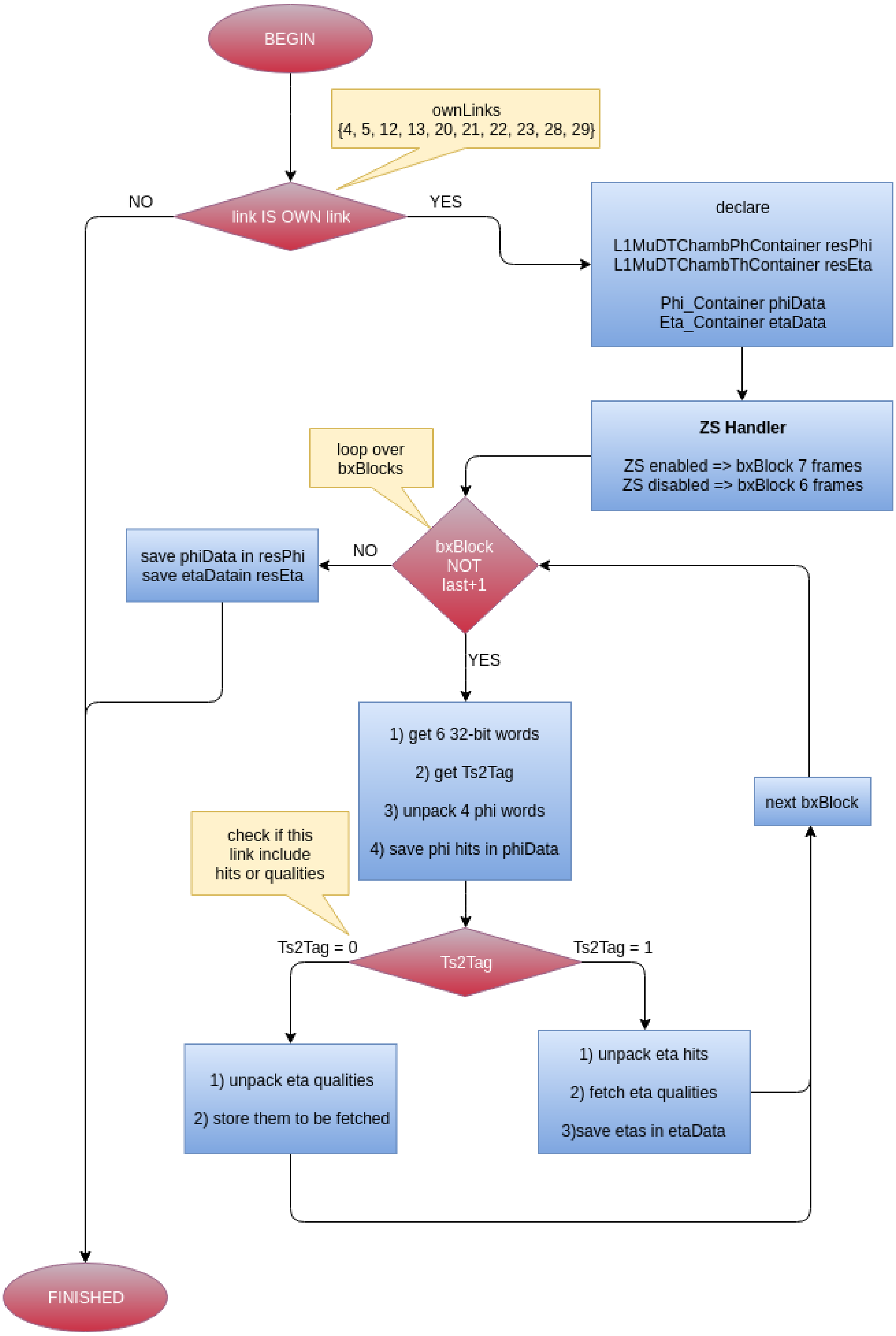
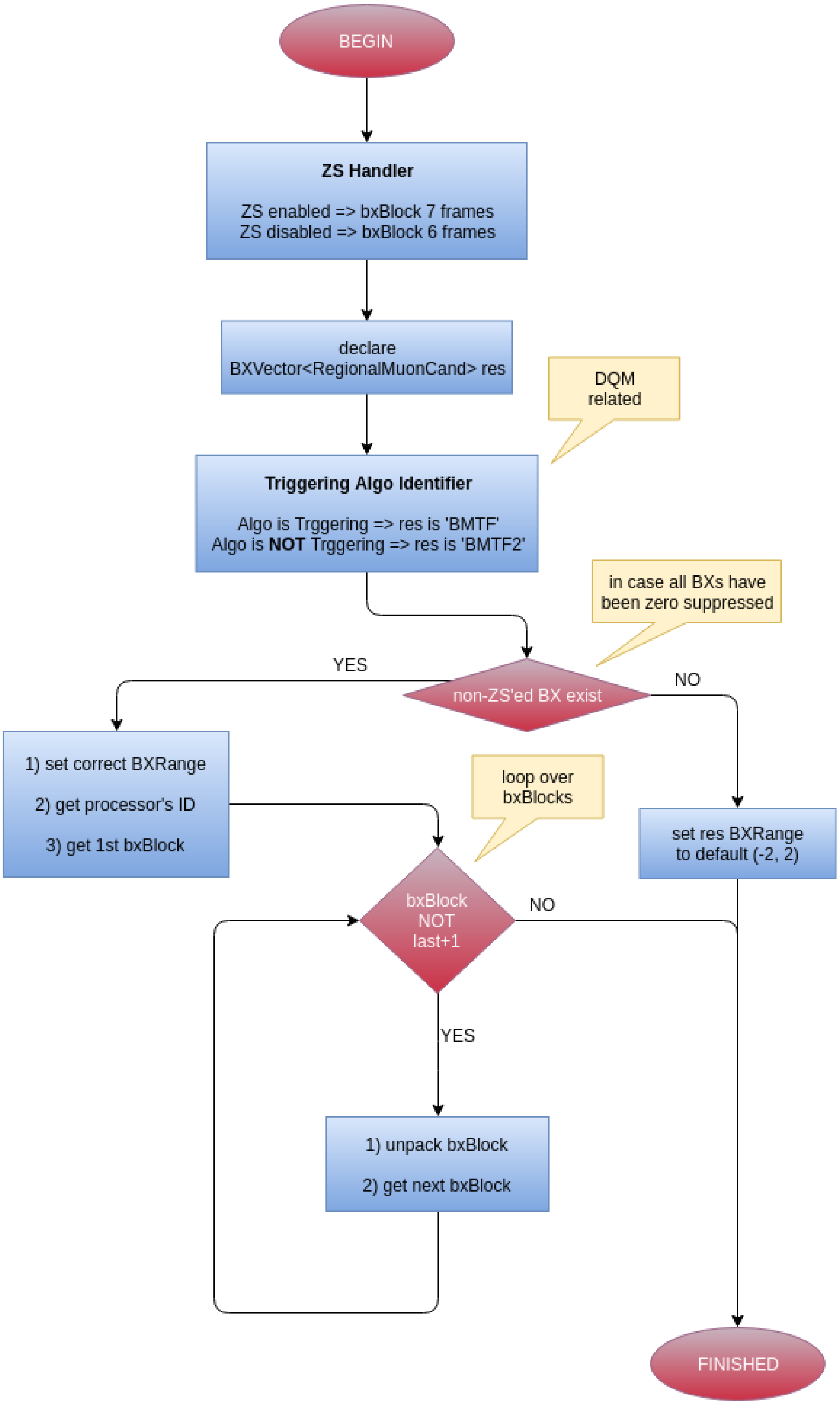


Figure 4.5: The Outputs' BMTF Unpacker Plugin.


BEGIN
ZS Handler
ZS enabled => bxBlock 7 frames
ZS disabled => bxBlock 6 frames
declare
BXVector<RegionalMuonCand> res
DQM related
Triggering Algo Identifier
Algo is Trggering => res is 'BMTF'
Algo is NOT Trggering => res is 'BMTF2'
in case all BXs have been zero suppressed
YES
non-ZS'ed BX exist
NO
1) set correct BXRange
2) get processor's ID
3) get 1st bxBlock
loop over bxBlocks
set res BXRange to default (-2, 2)
bxBlock NOT last+1
NO
YES
1) unpack bxBlock
2) get next bxBlock
FINISHED

regarding the ZS mode. Different handling needs to be performed on the BxBlocks in case ZS is activated. BxBlocks for Zero Suppressed data are one 32-bit word longer, as they contain a header for each BX which holds information about the suppression. Once all the BxBlocks, that are being included in the Block received by the plugin, have been extracted a loop over them is performed. The most important procedures in this loop are the unpacking of the most four ϕ and three η 32-bit words per BX, and the check of the ”Ts2Tag” flag which helps the plugin to identify if the link carries the position or the quality of the η hits[6]. The figure 4.12 displays the scheme of the BMTF payload for the input hits and according to that the plugin translates the binary information.

Figure 4.5 illustrates the outputs' plugin. The BMTF output consists of two links per processor. Each of them includes information for up to five bunch crosses and in each BX up to three L1 muons. Firstly, a similar Zero Suppression Handler is responsible to identify if the BMTF system operates with the ZS activated. Next in the sequence is the ”Triggering Algo” check. This is responsible to identify if the incoming link to the unpacker object contains the read out of the triggering algorithm or the secondary one that was running parasitically. In case the link comes from the triggering algorithm the unpacker creates and registers its product using the tag ”BMTF”, otherwise the unpacker will output its product tagged as ”BMTF2”. Last check before the binary to decimal translation is the identification of the non-Suppressed BxBlocks. This procedure is responsible to set the range of the BXVector which will be the output. If the link does not include non-Suppressed BX information then the range is set to a default value (5) in order to have BMTF output and uGMT input agreement. In the case non-Suppressed information found, a loop over the found BxBlocks is being performed to unpack the information and fill the BXVector with RegionalMuonCand objects. In figure 4.13 there is the payload scheme of BMTF's output, according to which the unpacking operation is performed.

Aside these two plugins which play the most important role in the unpacking procedure, BMTF owns also some other pieces of code to help the L1 supermodule to configure the required unpackers correctly. The most important of them is the ”BMTFSetup” and is responsible to create the appropriate unpackers per FED when it is called by the supermodule.

[6] BMTF receives information about its η inputs split into two consecutive links, one holds the position the next holds the quality.

4.2.2 The O2O Plugin

In order to operate correctly, the CMSSW modules require to be configured using a number of parameters. The Online-To-Offline[7] system configures the offline software and ensures that parameters which used both by software and firmware are identical. The parameters can be extracted from the L1 keys at configuration time. The O2O modules are responsible to store the required parameters for each system into the Conditions Database (CondDB), in order to be available for any future use.

The parameters can be anything that is used for calibration, masking or other configuration settings. These exist in the configuration keys and the O2O modules parse and store them. Below, figure 4.6, displays the main logic behind this mechanism. The CMSSW has the ”ParamsRcd” objects which are created by the O2O modules and are stored at the Conditions Database (CondDB). In the CondDB the different records that originate from different experiment eras are marked using a ”Tag” (is called ”GlobalTag” because it’s the same tag for all the systems included), and have a very specific Interval of Validity (IOV). The IOV is defined by the first time they are stored in the database until the next time the records are modified. Later, any CMSSW module that is designed appropriately using the GlobalTag is capable to make a request to the database and read the parameters in order to be configured.

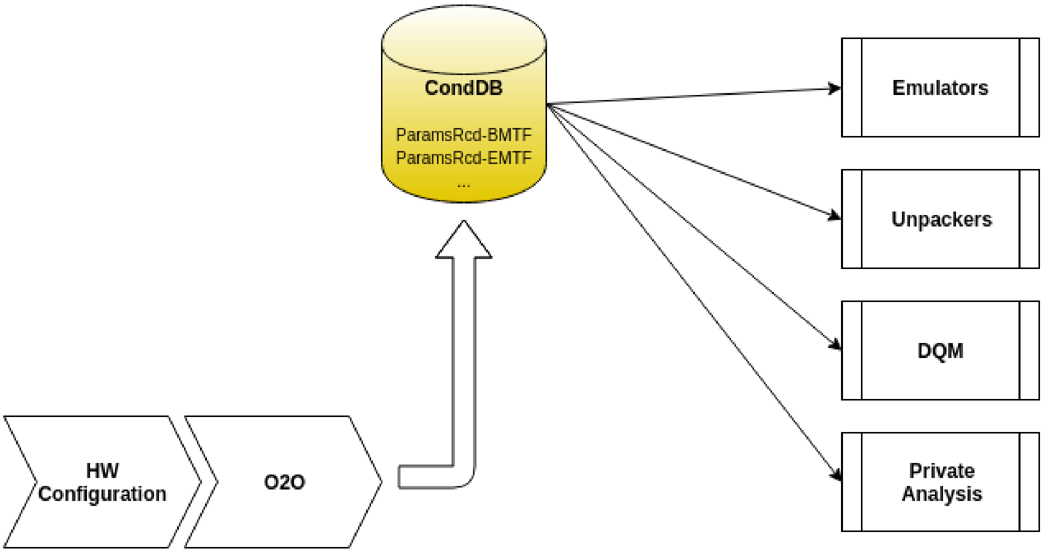


Figure 4.6: Schematic for the O2O Mechanism.

[7] The related packages to the L1 exist under the path ”L1TriggerConfig” and the BMTF module can be found under path ”L1TriggerConfig/L1TConfigProducers/src”.

For BMTF, the record is called ”L1TMuonBarrelParamsRcd” and contains information about the masking of the DTs in every run and several parameters that the Emulators need to operate correctly. As recently has been agreed it will include also the firmware version, because this can be used to distinguish different algorithm versions. The running algorithm version is necessary by the DQM to configure the online data to emulator comparison properly. For the tests, these days a short term solution has been used which doesn't include the O2O mechanism.

4.2.3 The DQM Plugin

The Data Quality Monitoring (DQM) is a web based tool. It presents plots in different folders organised per subsystem and per sub-detector. The advantage of having this tool is the flexibility that provides for quick troubleshooting in case of problems. DQM[8] requires each subsystem to provide a module that defines the plots to be shown in the subsystem's folder at the DQM GUI. Also, for the proper operation of the DQM, a correct configuration for the (globally used) comparison module of DQM must be provided. This configuration holds information about the firmware information and it must be propagated to the Emulators. This in general is being done using the O2O mechanism.

The BMTF DQM module defines plots for each hardware value that characterizes a Regional Muon Candidate (Pt, Eta, Phi, etc.). Also, plots exist for monitoring the Zero Suppression read-out rate reduction. For monitoring also the second algorithm (Kalman) in the BMTF FPGAs, a replica BMTF folder has been created in the DQM GUI (under the main BMTF path). This replica folder includes plots for the output of the secondary BMTF algorithm. Since two algorithms run in parallel, there has been the idea to add a new string monitorable into the BMTF folders. The new monitorable will display the name of algorithm for which plots will be presented in the folder[9]. However this is not yet implemented, it will be ready for Run 3 where a lot of people willing to try two algorithms in parallel for development purposes towards the Phase 2.

During the commissioning of the new algorithm the BMTF team implemented a new DQM plugin called ”BMTFAlgoSelector” (an EDProducer). Its purpose is to identify the triggering algorithm and place its output specifically at the DQM GUI (to be visible by the shifter). Figure 4.7 illustrates

[8] The related packages to the L1 exist under the paths ”DQM/L1TMonitor” and ”DQM/L1TMonitorClient”.

[9] Until now the distinction between the algorithms is being done only by looking at the spectra shapes and so is not easy to be done by the shifter

the selected solution for the DQM comparison to be performed correctly. It takes as input the two Regional Collections of Muons (coming from the two Emulators) and also the Raw Data in order to unpack and use the firmware version. The plugin is responsible to identify which algorithm is triggering during the run in hand (Kalman or Phase-1), and to help performing the DQM online comparison.

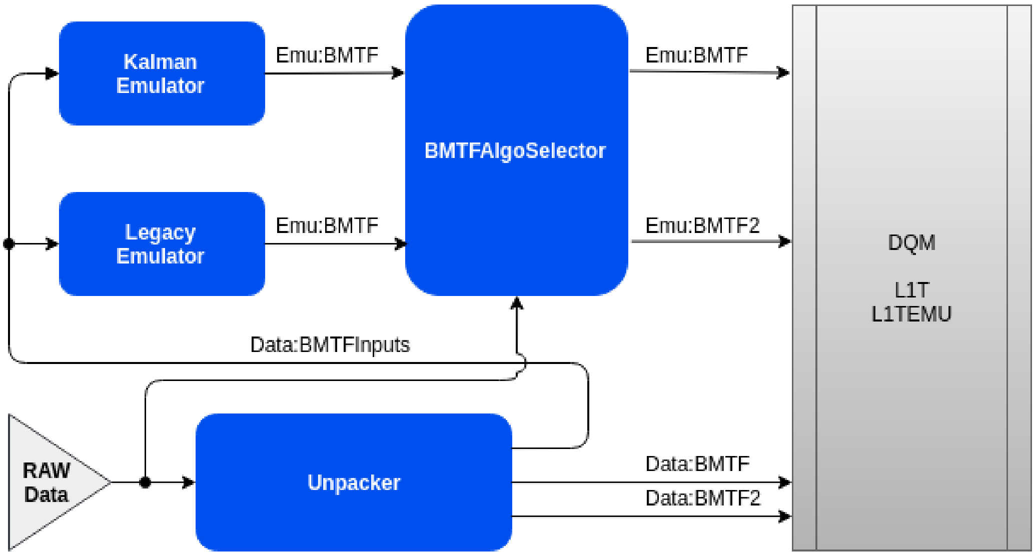


Figure 4.7: BMTF Modules in Cooperation at P5.

Figure 4.8 presents the logic behind the BMTFAlgoSelector. This plugin is the implementation of a 2x2 multiplexer. For plots included in the L1TEMU folder, the DQM compares the unpacked and the emulated RegionalMuonCands which have the same tag. For example, in the BMTF case, the ”Data:BMTF” muons will be compared to the ”Emu:BMTF” muons, and the same holds for the ”BMTF2” collections. The problem that this plugin solves is that the two BMTF emulators produce the same CMSSW object (vectors of L1 muons) and they tag them using the same tag, ”BMTF”. This raises the problem that the DQM has to make a choice which collection of muons to compare with which. The BMTFAlgoSelector makes the decision by looking at the ”algo-version”[10]. The firmware version is available into the unpacker, so the unpacker is capable of naming its outputs accordingly to which algorithm is triggering. The AlgoSelector handles the issue in the same fashion, unpacks the firmware version in order to use it and creates two copies of the emulators' collections tagging them in the same scheme like the unpacker (”BMTF” for the triggering and ”BMTF2” for the secondary).

[10] Algorithm version is part of the firmware version.

The algo-ver is agreed to be less that 0x95000160 (hex) for versions of the firmware in which the phase-1 algorithm is triggering, while values higher or equal to 0x95000160 have the versions in which the kalman algorithm is triggering. There is also a default output scheme (assuming that the phase-1 algorithm is triggering), which is used in case both output links are Zero Suppressed and the fw-ver information is not available (otherwise this module causes SEGFAULTs). Up to now this plugin is used having the ZS deactivated for the output link of the secondary algorithm. In the recent future the BMTFAlgoSelector will use the fw-ver that is stored in the CondDB.

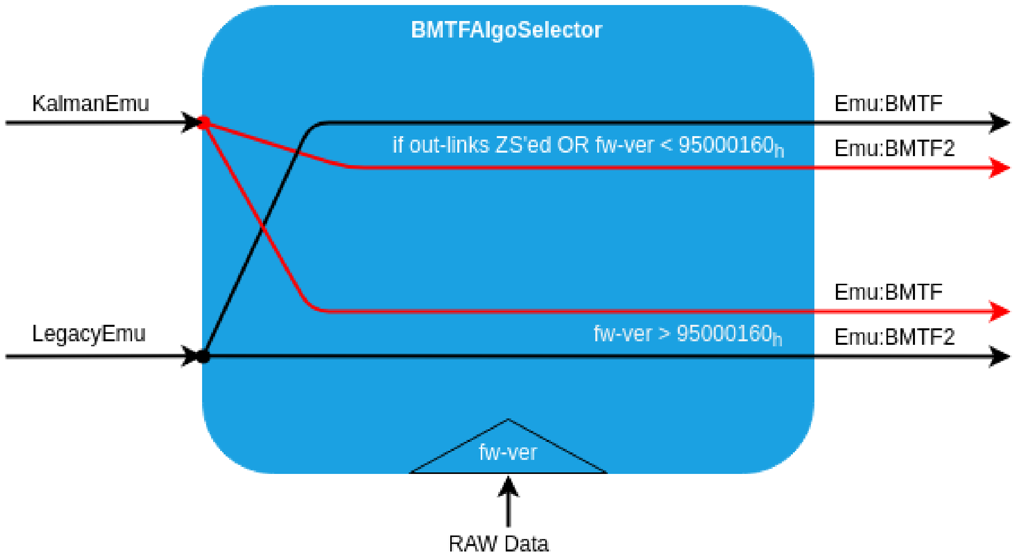


Figure 4.8: A closer look at the BMTFAlgoSelector.

4.3 Data Quality & Performance

4.3.1 The Emulator Modules

The current BMTF firmware (Fall 2018) contains in parallel two triggering algorithms, and both of them require a functional Emulator to be available in the CMSSW. Emulators[11] consume partially the output of the Unpacker (the BMTF input, phis and etas) and produces a regional collection of muons, which is a replica of what the BMTF firmware should have as output for this specific input (the BXVector of RegionalMuonCands). Firmware algorithms are required to agree with the emulator models more than 99% in order to be

[11] The packages for the L1 emulators exist under the path "L1Trigger", while for BMTF the emulators exist in the package "L1Trigger/L1TMuonBarrel"

used at CMS for data taking. This gives the flexibility of quick troubleshooting in case of faulty plots at the DQM folders at P5. The agreement level is tested using the module that is described in the next section, the Validator.

4.3.2 The Validation Module

The Validator is a private module, meaning that it's not uploaded into the CMSSW repositories at github, which exist in a private repository. It has been developed the last year and it is capable to validate simultaneously both the algorithms for a given run. It has been used extensively during the commissioning of the Kalman Algorithm at P5 and the tuning needed for making the Kalman emulator and the Kalman firmware to agree more than 99% as required. Validation plots for the muon variables "hwPt", "hwPt2"[12] and "hwPhi" are displayed for both algorithms in figure 4.9 for Phase-1 and in figure 4.10 for Kalman. For the plots, the cosmics run 321963 has been used. A more complete collection of these plots (all the interesting variables) can be found in the appendix.

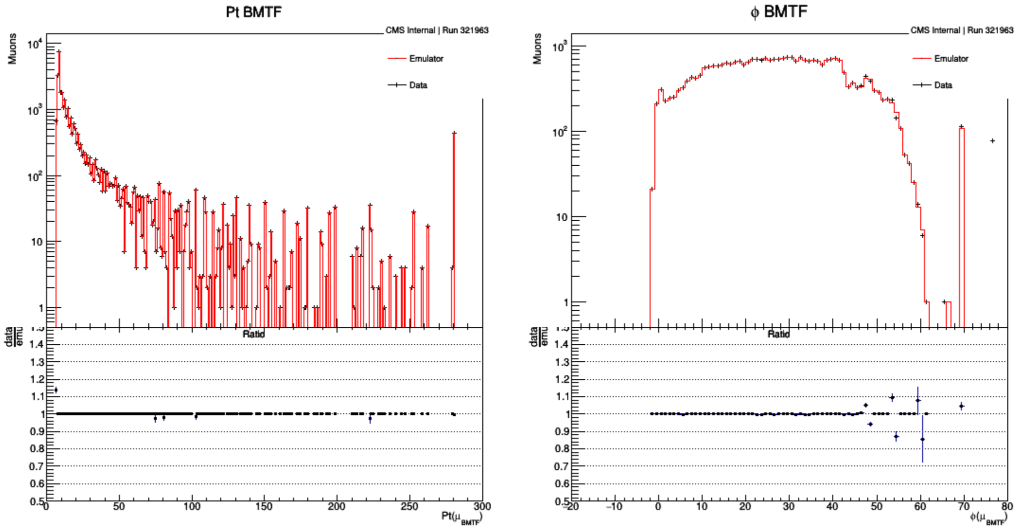


Figure 4.9: Validation Plots for the Phase-1 Algorithm.

Shown in figure 4.9 is the P_T (left) and ϕ (right) variables of the reconstructed BMTF muons. The emulator values are presented by the red histograms, while the firmware values are presented by the black marks. The graphs below the histograms are the ratios of the plotted above two hardware variables.

[12] Displaced Pt from the Kalman algorithm

The validator is a CMSSW module (an EDAnalyzer, see figure 4.11). It consumes output from the Unpacker and both Emulators. Its goal is to check for every RegionalMuonCandidate if the outputs of the Firmware (Unpacker) and the Emulator are the same (as it should be). In case of differences, mismatches are counted per variable and their percentage with respect to the sum of muons is calculated. The outputs of the module are three files. One ROOT file with several validation plots and two files (one per algorithm) which hold the printed out information for the events with mismatches.

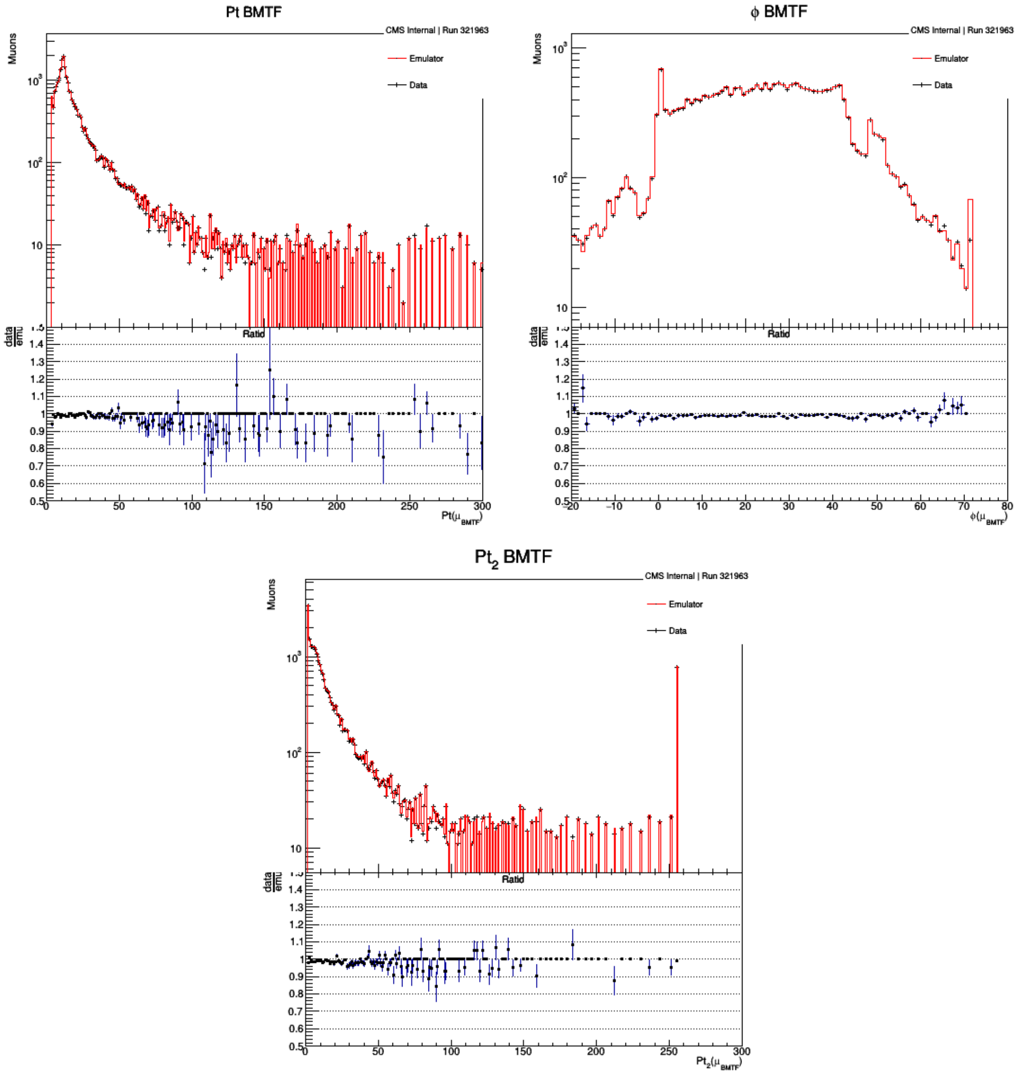


Figure 4.10: Validation Plots for the Kalman Algorithm.

Shown in figure 4.10 is the P_T (left), ϕ (right) and P_{T2} variables of the reconstructed BMTF muons coming from the Kalman algorithm. The variable P_{T2} is the vertex unconstrained measurement of the muon's momentum. The emulator values are presented by the red histograms, while the firmware values are presented by the black marks, and the graphs below the histograms are the ratios of the plotted above two hardware variables.

The algorithm of the validation module follows. Firstly an iteration over all the muons (data and emulator) is performed, in order to fill their variables into 1D histograms like above. The next is a muon-pair iteration per processor (iteration over data-emulator muon pairs). During this loop the 2D plots (look at the appendix) are being filled, and also the muon-pairs are being checked for mismatching variables. After the pair-loop, one more 2D histogram is filled with the number of muons in each BXVector[13]. Also the sizes are checked if differ (size mismatch check). Finally, the last step is to print into the two mismatch files information about the mismatching events[14]. Once all these five steps have been completed, a new event goes through the same procedure.

4.4 Complementary Software

4.4.1 The Packer Plugin

Another plugin, necessary for the Monte-Carlo (MC) Production, is the Packer. This module is complementary to the Unpacker and its functionality is completely the opposite. During MC Production, firstly events are generated and the related detector response is emulated. This is not being performed in binary format, so information is required to be ”packed” into binary and next to be FED to the CMS emulation chain (for example ”hits” are generated and they must be packed into binary to test the chain unpack-plus-emulate).

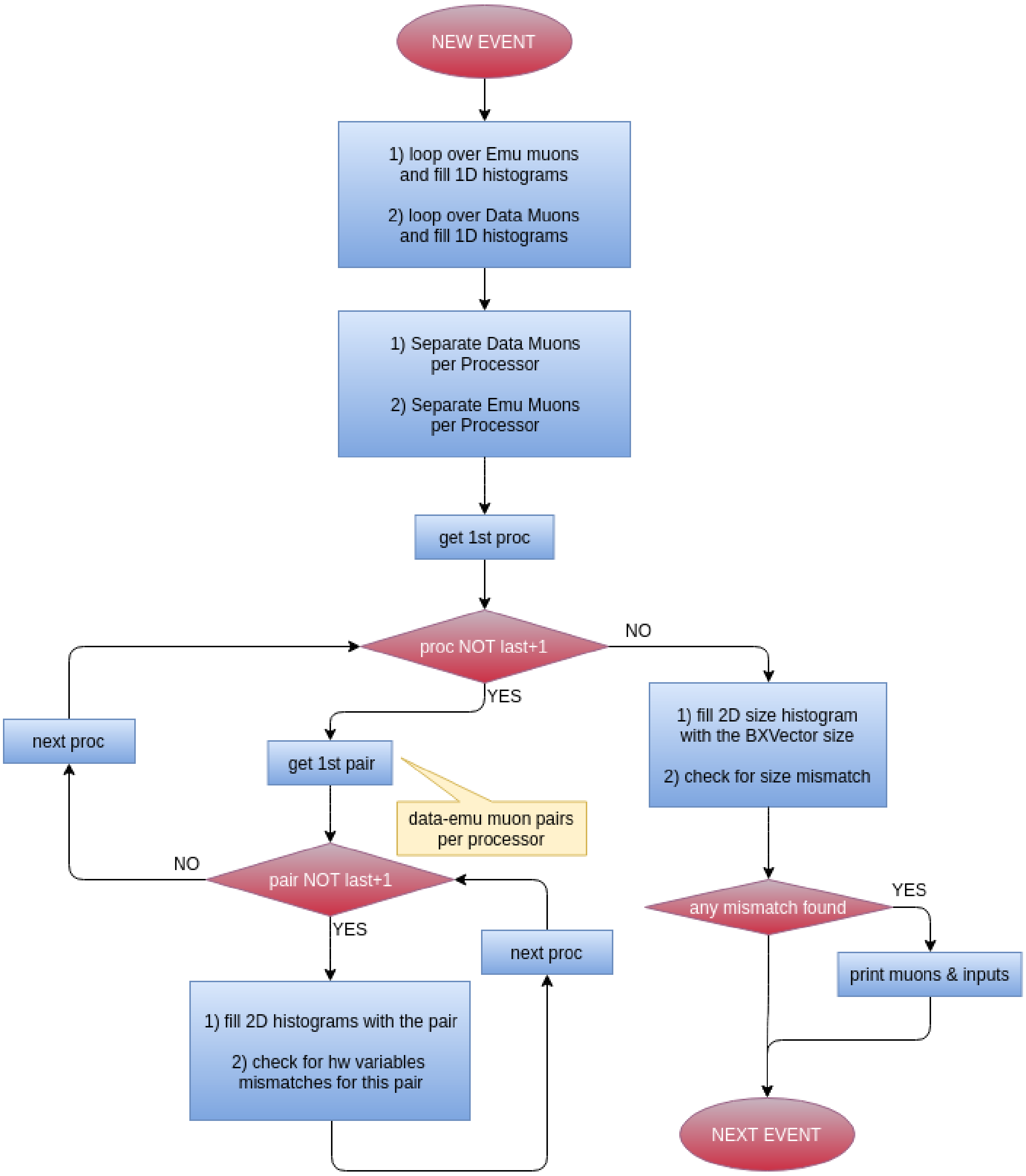
The Packer[15] (an EDProducer) follows the same logic with the Unpacker. Like for the unpacker case, the general L1 packer supermodule handles the packing procedure. The module is capable of handling plugins developed specifically for each subsystem, which are implementations of the Packer C++ class. The packers are objects responsible to deliver to the supermodule Block objects, which are put together and form the FEDRawDataCollection.

[13] One is the data BXVector and the other is the emulator BXVector.

[14] For each mismatching event is being printed the complete input hits, all the firmware output muons and all the emulator output muons.

[15] The related code exist exactly at the same place where is the unpacker's code.

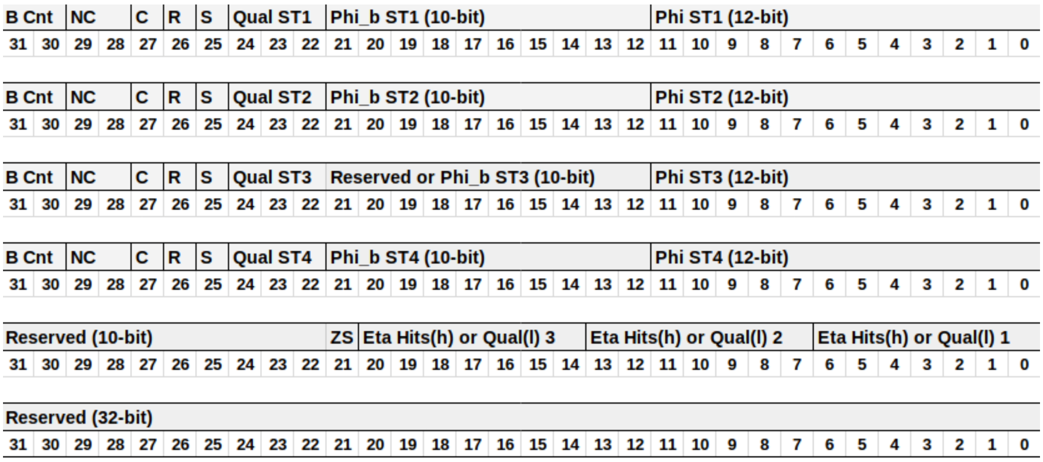
Figure 4.11: The Validator's Algorithm (per Event).


NEW EVENT
1) loop over Emu muons and fill 1D histograms
2) loop over Data Muons and fill 1D histograms
1) Separate Data Muons per Processor
2) Separate Emu Muons per Processor
get 1st proc
proc NOT last+1
NO
YES
next proc
get 1st pair
data-emu muon pairs per processor
1) fill 2D size histogram with the BXVector size
2) check for size mismatch
NO
pair NOT last+1
YES
next proc
1) fill 2D histograms with the pair
2) check for hw variables mismatches for this pair
any mismatch found
YES
print muons & inputs
NEXT EVENT

The BMTFPacker consumes exactly what the Unpacker produces, eta and phi inputs or Regional Muons, and translates them (produces) into binary information. For now, the BMTF team has provided software to pack only hits and RegionalMuonCands coming from the phase-1 algorithm. Once the Kalman algorithm becomes the main triggering system, an updated version of the Packer plugins must also be delivered to the CMSSW.

What follows is a brief description of the two "BMTFPacker" plugins which had been developed in August 2017. Also, a simple flowchart of the plugin for the inputs can be found at the end of the section. For the output plugin, the algorithm is simple enough to be described with words, later in this section.

The packer plugin for the input hits can be seen in figure 4.14. It is an implementation of the displayed algorithm in a per event and board fashion. The supermodule creates the appropriate number of packer objects and calls them one for each BMTF board. The BMTF Packer is a big link-based loop. If the link number corresponds to an input link then it's being processed, else the packer goes to the next link. For the links that correspond to inputs, firstly the phi hits from this link and processor are being identified. Using these hits up to four "phi words"[16] are being created.



B Cnt		NC		C	R	S	Qual ST1			Phi_b ST1 (10-bit)										Phi ST1 (12-bit)											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

B Cnt		NC		C	R	S	Qual ST2			Phi_b ST2 (10-bit)										Phi ST2 (12-bit)											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

B Cnt		NC		C	R	S	Qual ST3			Reserved or Phi_b ST3 (10-bit)										Phi ST3 (12-bit)											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

B Cnt		NC		C	R	S	Qual ST4			Phi_b ST4 (10-bit)										Phi ST4 (12-bit)											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved (10-bit)										ZS	Eta Hits(h) or Qual(l) 3							Eta Hits(h) or Qual(l) 2							Eta Hits(h) or Qual(l) 1						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved (32-bit)																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

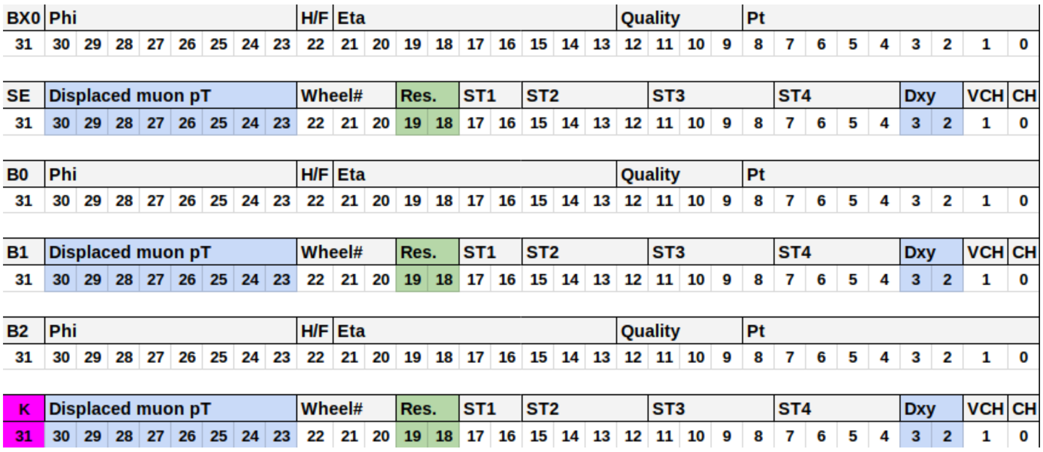
Figure 4.12: Inputs payload for the BMTF L1 subsystem.

For the eta hits the case is a little more complex. Their information is separated into two consequence links. A link check is being performed and if the link is even, the same procedure as for phi hits is following. For the even links, once the eta hits from the board and link in hand have been identified, one 32-bit word containing up to three "eta quality words" is being created

[16] see BMTF input payload, fig. 4.12

and the position of the hits are stored. For odd links the stored positions are being fetched and the corresponding 32-bit word is created. Before the next link in the loop, the Block object is being filled with one ”six 32-bit words” payload object.

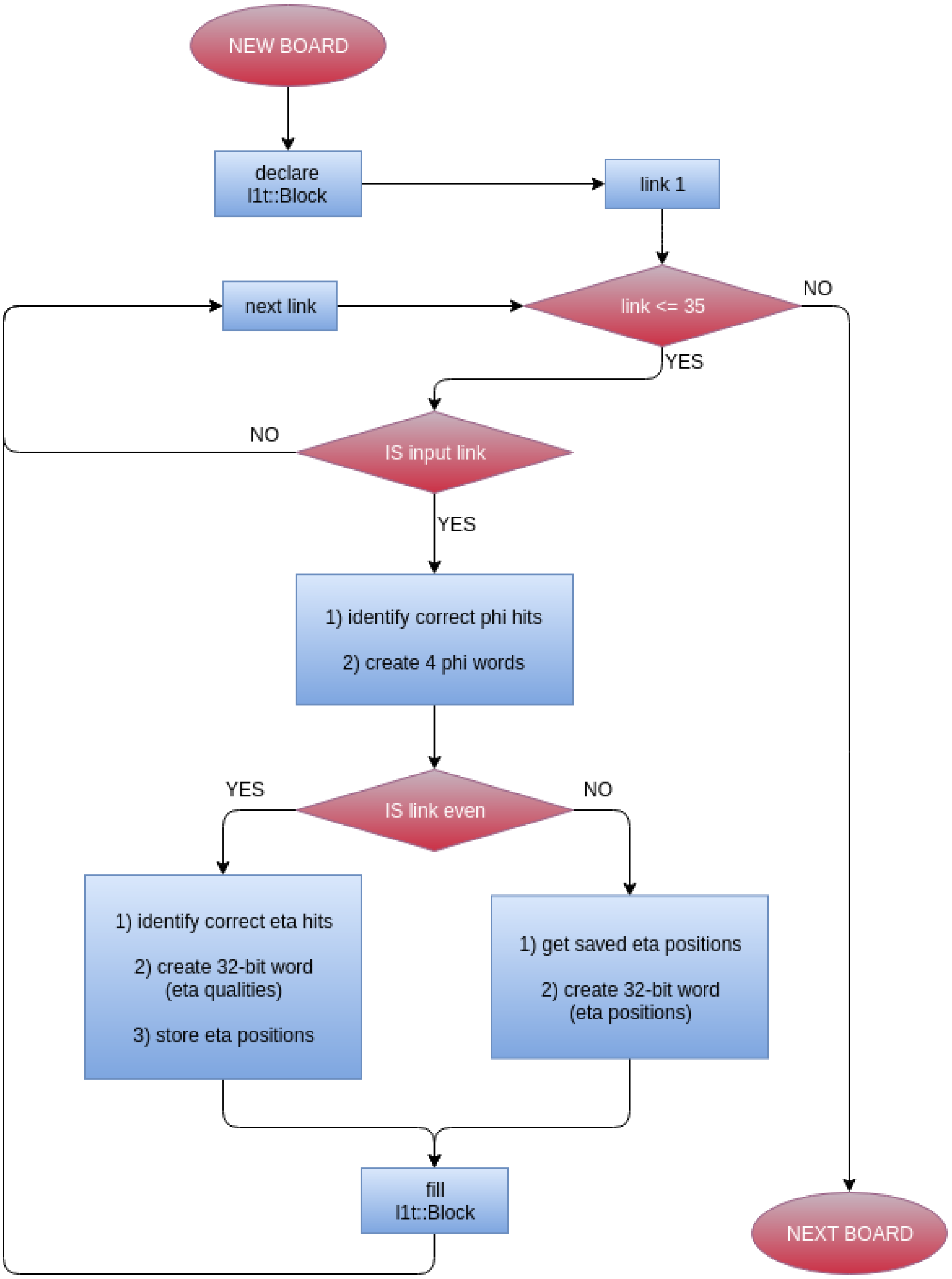
The plugin for the output Phase-1 muons follows the same logic as for inputs. However, the output link is only one per processor so there is no reason to have a link-loop. The packer objects that the supermodule handles per board are two. One related to the inputs and one to the output. Once the packer for the output is called for a specific board, the muons which had been found from this board are being identified. For each board the output payload contains up to three muons per bunch-crossing, whose hardware variables are being packed according to the figure 4.13.



BX0	Phi								H/F	Eta									Quality				Pt								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE	Displaced muon pT								Wheel#			Res.		ST1		ST2				ST3				ST4				Dxy		VCH	CH
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B0	Phi								H/F	Eta									Quality				Pt								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1	Displaced muon pT								Wheel#			Res.		ST1		ST2				ST3				ST4				Dxy		VCH	CH
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B2	Phi								H/F	Eta									Quality				Pt								
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K	Displaced muon pT								Wheel#			Res.		ST1		ST2				ST3				ST4				Dxy		VCH	CH
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 4.13: Output payload for the BMTF L1 subsystem.

Figure 4.14: The BMTF Inputs Packer's Algorithm (per Event & per Board).



Chapter 5

Expanding Soft-Opposite-Sign Analysis

5.1 Introduction

The work that is presented here targets to increase statistics for the SUSY Soft Opposite-Sign (SUSY-OS) analysis and expand the experimental reach for low ΔM scenarios where ΔM refers to the mass difference between the LSP[1] and the next to LSP. The final state of the SUSY-OS analysis comes from a chargino-neutralino pair production where those are assumed to decay into the virtual W^* and Z^* bosons respectively, producing also large amounts of missing energy. The process is displayed in figure 5.1. The analysis uses DoubleMuon triggers in L1, but in low ΔM cases one of the muons is more likely to fail firing the L1 trigger. This study's goal is to estimate the potential

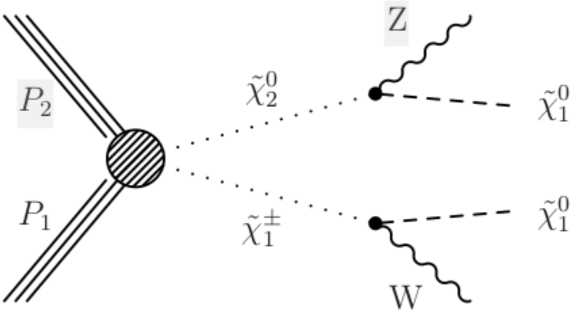


Figure 5.1: Chargino Neutralino Pair Production.

[1] LSP: Lightest Supersymmetric Particle.

variation of the signal region yields and increase of its significance when also using a SingleMuon trigger (calling it here ”SingleSoftMuon Trigger”) in addition to the existing trigger. SUSY-OS is a search for SUSY in compressed mass spectrum scenarios, involving E_T^{miss} and 2 soft leptons. Compressed-spectrum scenarios are called those which predict the LSP and the Next-to-LSP (NLSP) to be closely degenerate. This discussion focuses into the neutralino branch of the process, and also to mass differences between $\tilde{\chi}_2$ and $\tilde{\chi}_1$ lower than 20 GeV.

5.2 Triggers' Description

The study simulates the existing and the proposed triggers using cuts, the various cuts used are shown in Table 5.1.

Table 5.1: Triggers Summary

SUSY-OS Trigger	**SingleSoftMuon Trigger**
# muons L1 2 # muons RECO 2 $P_T^{L1}(\mu) > 3\text{GeV}$ $P_T^{RECO}(\mu) > 5\text{GeV}$ μ^{L1} Quality ≥ 12	# muons L1 1 # muons RECO 2 $P_T^{L1}(\mu) > 3\text{GeV}$ $P_T^{RECO}(\mu_1) > 5\text{GeV}$ $P_T^{RECO}(\mu_2) > 3\text{GeV}$ μ^{L1} Quality ≥ 12 $\lvert\eta^{L1}(\mu)\rvert < 1.5$
$P_T^{L1}(1jet) > 60\text{GeV}$ OR $P_T^{L1}(2jets) > 35\text{GeV}$ $\lvert\eta^{L1}(jet)\rvert < 2.5$	# Jet L1 1 $P_T^{L1}(jet) > 80\text{GeV}$ $\lvert\eta^{L1}(jet)\rvert < 2.5$
$E_T^{miss,L1} > 35\text{GeV}$	$E_T^{miss,L1} > 35\text{GeV}$

5.2.1 The SUSY-OS Trigger

This is a simulation for one of the triggers that SUSY-OS uses to collect data during 2018. Simulating the triggers for this study with cuts should serve 2 targets, firstly to have a selection similar (in philosophy) to the original trigger path, and secondly the selected events to be replicating the selection of the original analysis (so we don't fall out-of-context). The analysis focuses in events with at least two leptons which are final products of Z^*. To simulate the trigger two muons are required in both L1 and Offline reconstruction with

P_T above 3 GeV at L1 and above 5 GeV offline. Also, L1 muons are required to be of at least L1-Quality 12 (so that would be selected by the L1 system) and offline muons to be loose (since this is closer to the custom analysis muon tag).

For the non-leptonic part of the selection at least one hard Jet is required to boost the LSP[2]. So, this scenario would be selecting events including one L1 jet with P_T greater than 60 GeV or two L1 jets with P_T greater than 35 GeV. The signature of the LSP is expected to be missing transverse energy (E_T^{miss}), so it's also required L1 E_T^{miss} above 35 GeV. This way we simulate the E_T^{miss} part of the trigger too, with an extremely lowered threshold though due to limited statistics of our signal Monte Carlo.

5.2.2 The SingleSoftMuon Trigger

This is a simulation for the trigger which has been added in the Physics Menu of the CMS in 2018. This trigger accepts more background events (regarding the SUSY-OS analysis), however the group is relying on their hard selection cuts to clean the background efficiently. The concept is to gain efficiency at low ΔM scenarios, where the SUSY-OS trigger is incapable of triggering these events. The physical reasons could lay on the imbalance of the muons' P_T after the decay of the virtual Z^*, so the low momentum muon is recorded only by the Tracker and not by any of the L1 muon systems. This results into some inefficiency at very low ΔM values because of the L1 part of the trigger.

The SingleSoftMuon trigger (as we will be referring to this selection from here afterwards) accepts events in which at least one L1 muon with P_T above 3 GeV has been found and at least two offline muons with leading P_T greater than 5 GeV and sub-leading greater than 3 GeV. Also, the L1 muon is requested with at least L1-Quality 12 and inside the η region ± 1.5 (Table 5.1). The $|\eta|$ requirement is applied on the offline part of the original soft SingleMuon trigger, but we apply it here on the L1 muon as for simplicity reasons in this study we prefer to work only with events that include at least 2 muons[3]. Keeping this cut, even at the L1 side, makes the result more reliable.

The hadronic part of the trigger requires at least one L1 jet with P_T greater than 80 GeV while being inside the η region ± 2.5. Also, except the above requirements, the event is being accepted if it also includes L1 E_T^{miss}

[2] In the pair production scenario with low ΔM this jets come from initial state radiation.

[3] Otherwise, we would have to deal with cases outside the analysis scope and this is not the goal of this thesis.

above 35 GeV. Again these thresholds are lowered (like in the DoubleMuon case) for the same explained reasons.

5.3 Processed Datasets

5.3.1 Signal

For the signal a private Monte-Carlo production has been chosen. The signal generation step of the MC production has been performed using CMSSW configuration received from the analysis group. Four different compressed spectrum scenarios have been considered, $\Delta M = 3$, 5, 10 and 15 GeV. For ΔM 3, 5 and 10 GeV each generated sample includes 200000 events, while for ΔM 15 GeV a smaller sample of 20000 events has been generated in order to make a first estimation of this scenario as well. The simulation and reconstruction has been processed using CPU power and storage space provided by the Tier-2 computing cluster located at the University of Ioannina (Greece), under the CMSSW 90X environment. This private MC production consists of the following five steps.

- Signal Generation (GEN)
- Detector Simulation (SIM)
- Packing of the Information into RAW Format (DIGI2RAW)
- Unpacking and L1 Simulation (RAW2DIGI)
- Offline Reconstruction (RECO)

The software collection that has been used includes also Madgraph 5 and Pythia 8 for the GEN step and Geant 4 for the SIM step. All three of them are integrated into CMSSW 90X.

The signal samples have been generated without pile-up. For this reason modifying[4] the ”JetID” cuts which perform jet-cleaning was necessary. It’s important to highlight that this modification has been done only for the signal samples. Also, during the Monte-Carlo generation events have not been selected by any HLT path, because in the context of this study the event selection is handled by the triggers described in section 5.2.

[4] Instead of using the complete collection of cuts for the jets’ energy fractions (the Twiki can be found here [11] for 80X or here [12] for 94X.), the only cut selected is ”MuonEnergyFraction < 0.8”. Usage of the rest cuts was delivering awkward distributions for the jet multiplicity in the events resulting in wrong H_T calculations.

5.3.2 Background

For the final state which has been chosen the most significant background comes from the processes $t\bar{t} + jets$ and Drell-Yan (DY) including leptonic final states originated from Z and jets. The public CMS MC samples have been used for the study, however not all the processes which have been considered by SUSY-OS analysis have been processed. The background samples with low cross section have been omitted. A complete list of the samples which have been processed during this study are presented in table 5.2. The collection is not identical to the one that is used by the SUSY-OS, but the most significant processes, regarding the cross section, contribute in the study. So aside a systematic uncertainty (which is not being estimated) one can conclude into a primary result that would lead to a more extensive study later.

Table 5.2: Background Samples

Process	**Cross Section** (pb)
TTJets_SingleLeptFromT	182.2
TTJets_SingleLeptFromTbar	182.2
TTJets_DiLept	87.3
ST_t-channel_top_4f_inclusiveDecays	136.0
ST_t-channel_antitop_4f_inclusiveDecays	136.0
ST_tW_antitop_5f_inclusiveDecays	35.6
ST_tW_top_5f_inclusiveDecays	35.6
ST_s-channel_4f_leptonDecays	3.7
DYJetsToLL_M-5to50_HT-100to200	224.2
DYJetsToLL_M-5to50_HT-200to400	37.2
DYJetsToLL_M-5to50_HT-400to600	3.6
DYJetsToLL_M-5to50_HT-600toInf	1.1
DYJetsToLL_M-50_HT-70to100	209.6
DYJetsToLL_M-50_HT-100to200	177.0
DYJetsToLL_M-50_HT-200to400	54.3
DYJetsToLL_M-50_HT-400to600	7.0
WJetsToLNu_HT-100To200	1656.8
WJetsToLNu_HT-200To400	442.8
WJetsToLNu_HT-400To600	60.1
WJetsToLNu_HT-600To800	15.7
WJetsToLNu_HT-800To1200	6.5

5.3.3 Weighting of the MC Samples

The SUSY-OS analysis has used 2016 data corresponding to an integrated luminosity of $35.9fb^{-1}$ to perform their full study, for this reason the MC samples will be weighted to this number.

The cross sections of each process and the integrated luminosity have been used to weight the MC events and conclude into yields comparable to the analysis. Below, the expressions in 5.1, illustrate the procedure which is being followed for weighting the MC samples, assuming one has a histogram (h) for each process sample that includes "$\mathcal{N}$" number total of events.

$$\begin{gathered} \mathcal{N}' = \frac{\mathcal{N}}{\mathcal{N}_{MC}} \sigma_{Process} L_{int} \\ \mathcal{N} = \sum_{bin} h_{bin} \end{gathered} \tag{5.1}$$

In equation 5.1 "$\mathcal{N}$" is the number of events passed the selection cuts, $\mathcal{N}_{MC}$ is the MC events generated for each sample, L_{int} is the integrated luminosity, and $\sigma_{Process}$ is the cross section of each process[5]. The resulting number of weighted events is $\mathcal{N}'$ (the yield).

5.4 Primary Gain Estimations

5.4.1 Gain per ΔM

Using only signal MC samples to compare the two scenarios is not the best practice, however the target of this subsection is to make a primary estimate before one compares the performance of the triggers under the analysis context using also the background. The comparison in the plots below is between the triggered events of each case. One case is the SUSY-OS DoubleMuon trigger by itself, while the other case is the two triggers in an ”OR” combination.

The left plot of figure 5.2 shows clearly that the ”OR” combination of the selections results into more events for the signal. So, we illustrate here that extra events that were missed by the DoubleMuon trigger is possible to be collected by the SingleMuon. In the plot on the right the ratio of those two trigger simulated scenarios is presented (black histogram versus blue histogram). It is expected the gain to be weaker for greater ΔM separations. Greater ΔM means greater energy for the Z^* which concludes into both the

[5] All signal processes is assumed to have the same cross section ($183.69fb$) as suggested by the SUSY Twiki [13] for this specific process and ΔM points.

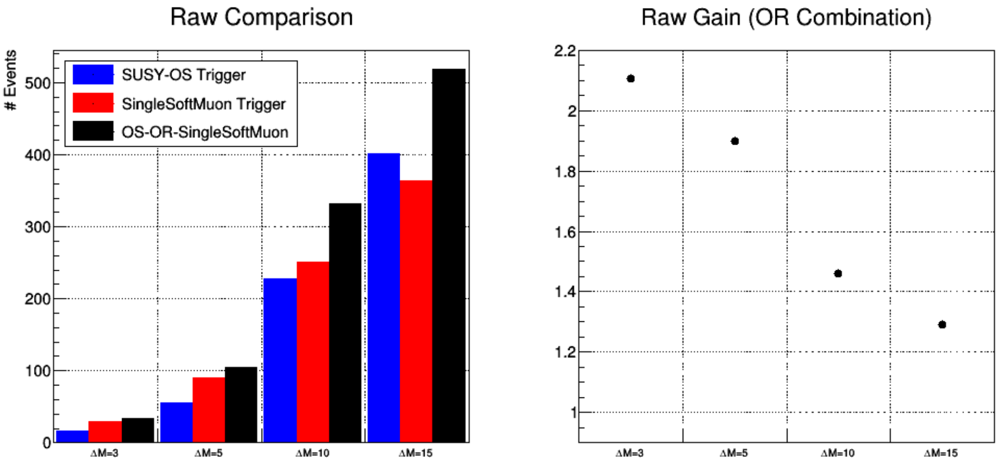


Figure 5.2: Left: Events accepted per Trigger per ΔM scenario. Right: Gain of combined triggers

muons to be reconstructed at the L1. Since the muons have more momentum the inefficiency of the L1 DoubleMuon part of the trigger at higher ΔM is not that significant.

5.4.2 Gain per P_T

Because of the more steep turn on of the offline efficiency for the muons, it's useful to study the accepted events per P_T for the two triggers. Their ratio gives important feedback regarding the P_T range where the assumed gain becomes insignificant. Figures 5.3 and 5.4 display the number of triggered events as a function of muon's P_T for all ΔM scenarios and for both triggering cases. The ratio per bin is also presented in the lower histograms and represents an estimate of the gain for each P_T bin. In the plots the expected trend is obvious. The higher the momentum of the muons, the less the gain because of the "OR" combination of the two triggers. So, as a conclusion, for higher momenta both trigger scenarios accept almost same number of events.

In summary, using only the signal samples there is an indication of profiting from using the Soft SingleMuon trigger. More events triggered in lower muon P_T values and lower ΔM points (exactly where the SUSY-OS analysis targets for 2018).

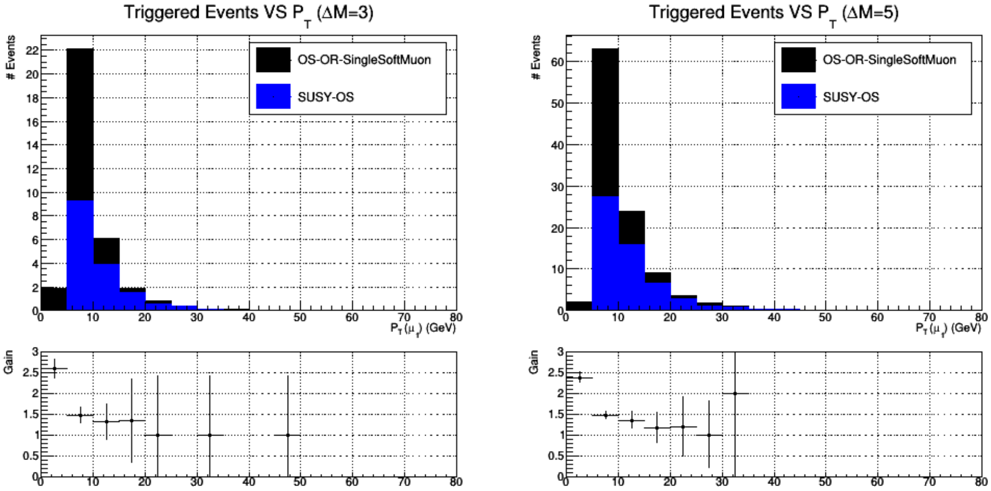


Figure 5.3: Leading P_T muons for $\Delta M = 3$(left) and $\Delta M = 5$(right).

5.5 Background Effect on the Gain

Since studying the signal samples have shown that introducing the Soft SingleMuon trigger increases the signal yields as expected, the full blown study is considered using both signal and background samples. For this study we will will use also offline cuts which are close to those used by the SUSY-OS analysis.

5.5.1 SUSY-OS Offline Analysis Cuts

Ultimately, the target is to compare the significance of the Signal Region (SR) under the two cases in question. The offline cuts that are being used by the analysis are of great importance for this reason, because they are designed to reject a great amount of events that come from the background processes. Table 5.3 summarizes the offline cuts which are used in the study. These cuts are applied on top of the events that have been selected by the two triggers. All the variables that are used for the cuts below, are offline variables.

However, the analysis has also 2 Control Regions (CRs) that each corresponds to the most important backgrounds TTJets and DYJets, the cuts defining the CRs are presented in table 5.4. In the following pages an estimation is also presented about how much these CRs would change after using also the SingleMuon trigger too.

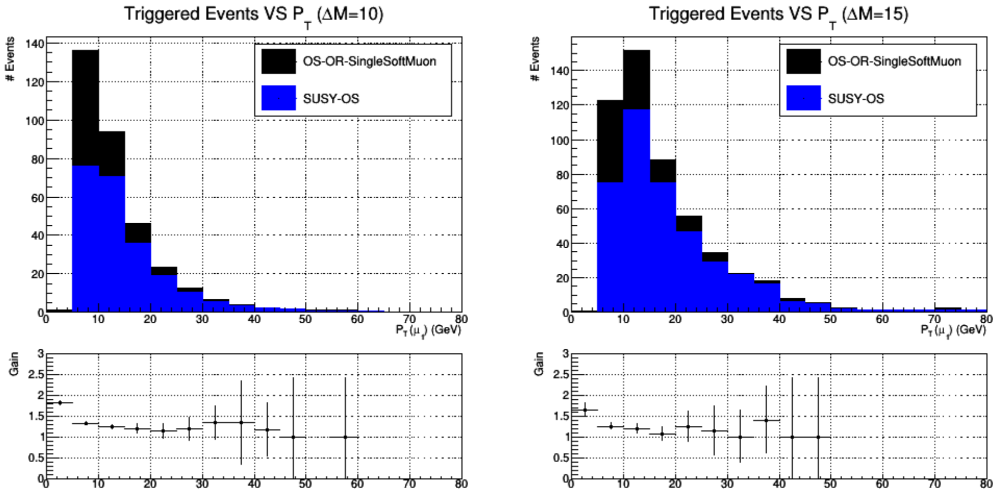


Figure 5.4: Leading P_T muons for $\Delta M = 10$(left) and $\Delta M = 15$(right).

Table 5.3: Offline Cuts Collection

$\#muons = 2$
$Q(\mu_1)Q(\mu_2) = -1$
$P_T(\mu\mu) > 3GeV$
$0 < M(\mu\mu) < 50GeV$
$E_T^{miss} > 125GeV$
$H_T > 100GeV$
$0.6 < E_T^{miss}/H_T < 1.4$
$\#jets \geq 1$
$P_T(jet1) > 25GeV$
$\lvert\eta(Jet1)\rvert < 2.4$

5.5.2 Gain per Background

An important thing to check is the effect on the background yields because of using the SingleMuon trigger. The accepted events is expected to be significantly higher since the number of the required L1 muons is now only one. The target is to calculate the number of events that are collected by each trigger and also passed the simulation of the Offline cuts too. The table 5.5 summarizes the results.

The ratios for the background yields are lower compared to the ratios presented in figure 5.2. For the important processes of $t\bar{t} + jets$ and $DY + jets$ the estimated gains variate up to 40% and 60% respectively in this custom and non-optimized SR. Also, important yield gain seems to be the case for

Table 5.4: Control Region Specific Cuts.

DYJets Region	**TTJets Region**
$P_T(\mu_1) < 20GeV$	$\#jets \geq 1$
	$P_T(jet1) > 40$

the $W + jets$ background. Removing the requirement for a second muon at the L1 more events that have one good L1 muon and possibly one or more pile-up muons is possible to have been collected, making the $W+jets$ process to fake the signal in more cases. However, the analysis could possibly handle this raise using other cuts that haven't been simulated here or by optimizing better the SR.

For the most significant background samples, we have the analysis control regions $t\bar{t} + jets$ and $DY + jets$, and the offline selection there is different as presented in table 5.4. The plots in figures 5.5 and 5.6 present the distributions of the highest P_T muon for those processes in these CR selections. The shapes of the distributions have been compared to their analysis counterparts and agree[6]. Again, one needs to keep in mind that there are omitted cuts here that exist in the analysis CRs, so the yields are expected different and are not presented. At this stage of the study any possible change in the shape of these distributions is the information that is considered useful.

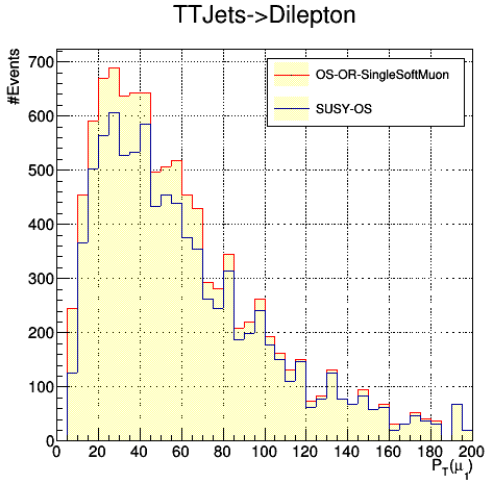


Figure 5.5: Accepted Events for $t\bar{t}$-Jets Control Region.

[6] The analysis note of the SUSY-OS analysis can be found here [14]

Table 5.5: Triggered Events and Rates per Background

Process	**SUSY-OS**	**SingleSoftMuon**	**Ratio**
TTJets_SingleLeptFromT	3564.35	4839.67	1.36
TTJets_SingleLeptFromTbar	4708.87	6485.6	1.38
TTJets_DiLept	8771.66	10156.1	1.16
ST_s-channel_4f_leptonDecays	30.3907	38.539	1.27
ST_t-channel_top_4f_inclusiveDecays	268.571	284.849	1.06
ST_t-channel_antitop_4f_inclusiveDecays	162.771	236.017	1.45
ST_tW_top_5f_inclusiveDecays	0	0	0.00
ST_tW_antitop_5f_inclusiveDecays	0	0	0.00
DYJetsToLL_M-5to50_HT-100to200	13.4146	13.4146	1.00
DYJetsToLL_M-5to50_HT-200to400	8.9032	11.129	1.25
DYJetsToLL_M-5to50_HT-400to600	5.57084	7.07068	1.27
DYJetsToLL_M-5to50_HT-600toInf	10.0206	11.3657	1.13
DYJetsToLL_M-50_HT-70to100	0	0	0.00
DYJetsToLL_M-50_HT-100to200	52.9639	84.7422	1.60
DYJetsToLL_M-50_HT-200to400	246.886	295.614	1.20
DYJetsToLL_M-50_HT-400to600	75.6051	87.301	1.15
WJetsToLNu_HT-100To200	297.397	594.795	2.00
WJetsToLNu_HT-200To400	821.32	1165.74	1.42
WJetsToLNu_HT-400To600	273.508	388.67	1.42
WJetsToLNu_HT-600To800	131.882	169.563	1.29
WJetsToLNu_HT-800To1200	70.8411	99.4872	1.40

5.5.3 Significance

The figure of merit (*FoM*) which is used here is the significance of signal's excess against the background (similar to one of the *FoM*s that SUSY-OS analysis uses). The calculation is performed for all the 4 ΔM scenarios, with and without including the Single Muon trigger. What follows is how the significance is defined. This same formula is used by the study.

$$sig = \frac{s}{\sqrt{s+b}} \tag{5.2}$$

s: signal events

b: all background events

Table 5.6 shows the calculated significance for each combination of triggers and ΔM scenario. The errors are only statistical and the systematic error because of the missing background samples has been omitted (as this

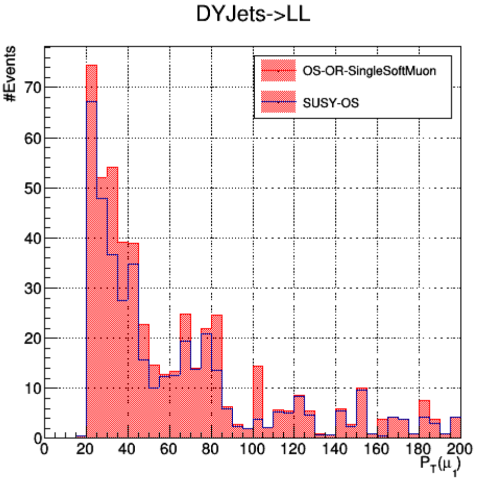


Figure 5.6: Accepted Events for DY-Jets Control Region.

Table 5.6: Significance per ΔM and per Trigger Scenarios

ΔM Scenario	**SUSY-OS**	**OS-OR-SingleSoftMuon**	**Ratio**
3	0.0357 ± 0.0038	0.0833 ± 0.0055	2.33
5	0.143 ± 0.008	0.247 ± 0.009	1.74
10	0.468 ± 0.013	0.673 ± 0.015	1.44
15	0.599 ± 0.048	0.781 ± 0.051	1.30

falls outside the context of this thesis). The resulting ratios of significance with and without the SingleMuon trigger are similar to the values that are presented in the right plot of the figure 5.2.

5.6 Conclusion

The results suggest that introducing the Soft SingleMuon trigger in logical OR with the existing DiMuon trigger can potentially lead to significant gain of signal events at lower ΔM . For this reason a complete study using the analysis framework and also emulating real HLT path is interesting to be done. There are enough indications, out of this study, that a significant boost to the sensitivity towards low ΔM scenarios is possible to be achieved using this trigger.

Chapter 6

Epilogue

Instead of an epilogue, here is presented a summary of the contributions described through the document. Because of the new Kalman algorithm for the Barrel Muon Track Finder, and the implication which have been risen during the development, testing and commissioning, a lot of changes have been made on the software. The new algorithm has been studied in cooperation with the UCLA (USA) and UOA (Greece) CMS teams, and now is fully commissioned for run 3.

- The BMTF Packer, which is software that is used in the Monte Carlo procedure, has been developed for the Phase-1 algorithm of the BMTF.
- The BMTF Unpacker, which is software that is used for data decoding from binary format to C++ objects, has been upgraded. Added new functionality in order to be capable of handling (a)the Zero Suppression or the readout record, (b)the Kalman algorithm output and (c)the requirements of dynamical switching (or selecting) the triggering algorithm between the Phase-1 and the Kalman algorithms.
- Developed a new DQM plugin, called ”BMTFAlgoSelector”, which has been of central role for the proper operation of the system after the second algorithm has been added in parallel in the BMTF FPGAs.
- Management of the BMTF software in the online repositories of the CMSSW keeping all the developments and the software at P5 up to date.
- A new validator for the BMTF system, capable of validating both the algorithms simultaneously, has been developed in order to fully commission the new Kalman firmware with the required agreement level of more than 99% with respect to the Kalman emulator.

- The Online Software of the BMTF (BMTF Swatch Cell) has been modified slightly to meet the latest requirements of the L1 operation group.
- Development has been done to the Online Software in order to (a)add support for two running algorithms in parallel and (b)configure the BMTF boards only with the requested firmware using a parameter in the L1 configuration keys.
- A physics study in the context of the SUSY Soft Opposite Sign Analysis has been performed. The purpose of the study is to estimate the possible gain of the analysis yields under the case of using also a SignleMuon trigger in logical OR with the current DiMuon trigger which is used by the analysis.

Appendix A

Derivation of the Kalman Filter Equations

The Kalman filter is designed to estimate the values of the incoming data. The more data it processes the better is its estimation for the next incoming signal. Its objective is to minimize the mean squared error between the estimated data and the actual ones (in this context is assumed that all the measurement errors follow Gaussian distributions). This fact allows to have a χ^2 merit function in order to evaluate the success of the algorithm achieving its target.

Towards the rest section, $\widetilde{\chi_k}$ will be the prior estimation, χ_k' will be the updated estimation, $z_k = f(x_k) = H\chi_k$ will be the inserted measurement and its representation in the abstract χ-space. The general rule that is being followed throughout the section is a "tilde" ($\tilde{\alpha}$) for prior values, a "prime" (α') for updated estimations and no punctuation (α) for the representation related to measurements. Firstly, the calculation minimizes the χ^2 and via the minimization derives minimization conditions. It occurs the Kalman equations to be these minimization conditions.

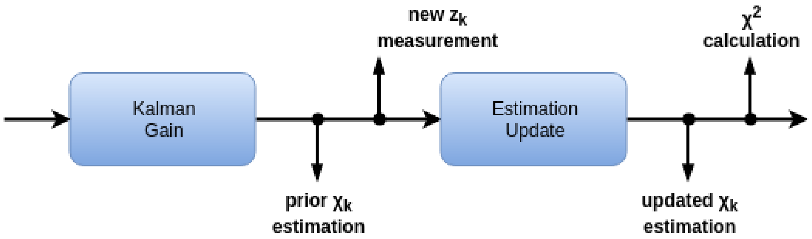


Figure A.1: Supportive image for the χ^2 calculation.

Figure A.1 explains at which point of the Kalman loop the calculation is performed, and also displays the variables that are being handled. The signal is assumed to be of the form that displayed in equation A.1, where $\mathbf{\Phi}$ is the theoretical transition matrix from one signal to the next and w_k is the white noise term. Equation A.2 formulates the signal in the observation native space. The matrix H is a noiseless transformation between z-space and χ-space, while u_k is the observation error. Errors w and u assumed to be non-correlated ($E[w_k u_k^\top] = 0$).

$$\chi_{k+1} = \Phi \chi_k + w_k \tag{A.1}$$

$$z_k = H \chi_k + u_k \tag{A.2}$$

A χ^2 function under the Kalman Filter context can be written down as in equation A.3. For the "prior" term of the χ^2 the mean squared error corresponds to the estimated covariance $\widetilde{P_k} = E[(\chi_k' - \widetilde{\chi_k})(\chi_k' - \widetilde{\chi_k})^\top]$ (also uncorrelated to w_k and u_k noise models). The first term of equation A.3 is related to the error between the prior and the updated estimates. The "updated" term is related to the error because of the new measurement. The mean squared error corresponds to the observation noise covariance $R = E[u_k u_k^\top]$.

$$\begin{gathered}\chi^2 = \sum_k \frac{(z_k - f(x_k'))^2}{\sigma_k^2} = \chi^2{}_{\substack{prior\\estimate}} + \chi^2_{\substack{updated\\estimate}} \\ = (\chi_k' - \widetilde{\chi_k})^\top P_k^{-1} (\chi_k' - \widetilde{\chi_k}) + [z_k - f(\chi_k')]^\top R^{-1} [z_k - f(\chi_k')]\end{gathered} \tag{A.3}$$

To derive the update equation for the signal's estimate, one needs to minimize χ^2. Firstly, the derivative with respect to χ_k' is performed and then the Kalman Gain and the χ-update equation are extracted.

$$\begin{aligned} 0 &= \frac{d\chi^2}{d\chi_k'} \\ &= 2\left\{ P_k^{-1}(\chi_k' - \widetilde{\chi_k}) - \frac{df^\top}{d\chi_k'} R^{-1} [z_k - f(\chi_k')] \right\} \end{aligned} \tag{A.4}$$

The Taylor expansion of function $f(\chi_k')$ around the prior estimation $\widetilde{\chi_k}$ can be written as follows.

$$f(\widetilde{\chi_k} + \Delta\chi_k) = f(\widetilde{\chi_k}) + \frac{df(\widetilde{\chi_k})}{d\chi_k'} \Delta\chi_k, \quad \Delta\chi_k = \chi_k' - \widetilde{\chi_k} \tag{A.5}$$

Substituting the expansion in expression A.4 and also using the fact that the derivative of the selected function $f(\chi_k)$ is just H (since $z_k = H\chi_k$), one can obtain the equation 3.1 if recognises the Kalman Gain G_k to have the form A.7.

$$\begin{aligned} \Delta\chi_k &= (P_k^{-1} + H^\top R^{-1} H)^{-1} H^\top R^{-1} [z_k - f(\chi_k')] \\ &\Rightarrow \boxed{\chi_k' = \widetilde{\chi_k} + G_k(z_k - H\widetilde{\chi_k})} \end{aligned} \tag{A.6}$$

$$G_k = (P_k^{-1} + H^\top R^{-1} H)^{-1} H^\top R^{-1} \tag{A.7}$$

Using the equation inside the box of A.6, and also A.2 one can conclude to the form below.

$$\chi_k' = \widetilde{\chi_k} + G_k(H\chi_k + u_k - H\widetilde{\chi_k}) \tag{A.8}$$

This expression can be used for the calculation of the error between the updated estimate and the signal. Its covariance is written as follows.

$$\begin{aligned} P_k' &= E[(\chi_k - \chi_k')(\chi_k - \chi_k')^\top] \\ &= E\left\{[(\mathbb{1} - G_k H)(\chi_k - \widetilde{\chi_k}) - G_k u_k][(\mathbb{1} - G_k H)(\chi_k - \widetilde{\chi_k}) - G_k u_k]^\top\right\} \\ &= (\mathbb{1} - G_k H)\widetilde{P_k}(\mathbb{1} - G_k H)^\top + G_k R G_k^\top \end{aligned} \tag{A.9}$$

Kalman filter provides estimations of the signal and the covariant matrix which are the optimal ones. This means P_k' will be minimum once it will be updated using the Kalman Gain. Hence, the Kalman Gain can be derived by minimizing this covariance.

$$\begin{aligned} 0 &= \frac{dP_k'}{dG_k} \\ &= \frac{d(\mathbb{1} - G_k H)}{dG_k}\widetilde{P_k}(\mathbb{1} - G_k H)^\top + (\mathbb{1} - G_k H)\widetilde{P_k}\frac{d(\mathbb{1} - G_k H)^\top}{dG_k} + 2G_k R \\ &= -2\left\{(\mathbb{1} - G_k H)\widetilde{P_k}H^\top + G_k R\right\} \\ &\Rightarrow \boxed{G_k = \widetilde{P_k}H^\top(R + H\widetilde{P_k}H^\top)^{-1}} \end{aligned} \tag{A.10}$$

The equation in the box, at the last line of A.10, is the equation 3.4 of Chapter 3. Substituting it to the expression A.9 and after some matrix manipulation one concludes into equation 3.2 of Chapter 3.

$$\begin{aligned} P_k' &= (\mathbb{1} - G_k H)\widetilde{P_k}(\mathbb{1} - G_k H)^\top + G_k R G_k^\top \\ &= (\mathbb{1} - G_k H)\widetilde{P_k} + \left(\widetilde{P_k}H^\top G_k^\top - G_k(H\widetilde{P_k}H^\top + R)G_k^\top\right) \\ G_k &\rightarrow \widetilde{P_k}H^\top(R + H\widetilde{P_k}H^\top)^{-1} \\ &\Rightarrow \boxed{P_k' = (\mathbb{1} - G_k H)\widetilde{P_k}} \end{aligned} \tag{A.11}$$

Concluding one needs also to project the estimations to the next input signal. For the state vector χ_k this can be dome using the transition matrix $\boldsymbol{\Phi}$.

$$\boxed{\chi_{k+1} = \Phi\chi_k} \tag{A.12}$$

For the covariance matrix, is necessary to write down the expectation value for the $k+1$ signal and then to relate it with the updated P_k.

$$\begin{aligned} e_{k+1} = \chi_{k+1} - \widetilde{\chi_{k+1}} &= (\Phi\chi_k + w_k) - \Phi\widetilde{\chi_k} \\ &= \Phi(\chi_k - \widetilde{\chi_k}) + w_k \\ \Rightarrow e_{k+1} &= \Phi e_k + w_k \end{aligned} \tag{A.13}$$

Finaly expression A.13 can be used to calculate the projection of the covariance matrix.

$$\begin{aligned} P_{k+1} = E[e_{k+1}e_{k+1}^\top] &= E[(\Phi e_k + w_k)(\Phi e_k + w_k)^\top] \\ &= \Phi E[e_k e_k^\top]\Phi^\top + E[w_k w_k^\top] \\ &\Rightarrow \boxed{P_{k+1} = \Phi P_k \Phi^\top + Q} \end{aligned} \tag{A.14}$$

Appendix B

The Configuration and Run Settings Keys

The keys for BMTF subsystem are xml files which include information that is required by the SWATCH cell in order to initialize and handle the MP7 boards. The files are separated in 4 categories, ”Hardware”, ”Infra”, ”Algo” and ”Run Settings”. Two of those categories, Infra and Run Settings, consist of keys for either MP7 or AMC13 boards. The other two categories include one xml file each, since the required information for both boards is included in the same key.

Beginning from the hardware key, this is the most important. It describes the the C++ objects which need to be implemented by SWATCH in order to initialize the system and also to communicate with the boards. The two lines below are significant, first holds the system's name and the second is the C++ class which will be used for the system.

```
<system id=”bmtf”>
<creator>swatch::bmtfswatchcell::BMTFSystem</creator>
```

Apart from the lines above, there are four blocks of code in the hardware keys which describe entirely the system, again using the same logic as the above lines. The blocks hold information about the attributes of object that is part of the ”BMTFSystem”. The crates don't need an object to be described, the are filled as attributes to the corresponding boards, so the below two lines define a crate.

```
<crate id=”Bottom_Crate”>
  <description>Bottom Schroff crate</description>
  <location>Point5 , S1D10−43</location>
</crate>
```

Significant is the description of the MP7 boards. For the cell the boards are ”processors”, and for the BMTF cell are ”BMTFProcessors” C++ objects. The below lines (except the port tags which can be customized) required in order the cell to communicate with a board and to initialize it. Two of these tags are of great importance, the ”uri” which is the network address of the board and the ”address-table” which is a collection of other xml files that maps the available registers to be handles by the cell.

```
<processor id="wedge07">
  <creator>bmtfswatchcell::BMTFProcessor</creator>
  <hw-type>MP7 XE</hw-type>
  <role>procMP7</role>
  <uri>chtcp-2.0://bridge-bmtf.cms:10203?target=amc-s1d03
 -09-02:50001</uri>
  <address-table>file:///opt/xdaq/etc/bmtf/addr_table/
 framework/mp7xe_infra.xml</address-table>
  <crate>Bottom_Crate</crate>
  <slot>2</slot>
  <rx-port pid="[00:06]" name="Rx[00:06]"/>
  <rx-port pid="[08:14]" name="Rx[08:14]"/>
  <rx-port pid="[16:30]" name="Rx[16:30]"/>
  <rx-port pid="[32:36]" name="Rx[32:36]"/>
  <tx-port pid="61" name="Kalman_Tx_61"/>
  <tx-port pid="62" name="BMTF_Tx_62"/>
</processor>
```

Similar logic is followed also for the description of the two AMC13s. Two uri tags and two address-tables because AMC13 has two FPGAs fitted on the board. One more tag here that is important is the ”fed-id”. BMTF owns two feds, 1376 and 1377. Both consist of links coming from half the barrel, if they swap between the two AMC13s then the read-out will be swapped as well, which means bad quality of data.

```
<daqttc-mgr id="AMC13_TOP">
  <creator>swatch::amc13::AMC13Manager</creator>
  <role>daqttc</role>
  <crate>Top_Crate</crate>
  <slot>13</slot>
  <uri id="t1">chtcp-2.0://bridge-bmtf.cms:10203?target=amc-
 s1d03-17-13-t1:50001</uri>
  <uri id="t2">chtcp-2.0://bridge-bmtf.cms:10203?target=amc-
 s1d03-17-13-t2:50001</uri>
```

```
    <address-table id=”t1”>file:///opt/cactus/etc/amc13/
   AMC13XG_T1.xml</address-table>
    <address-table id=”t2”>file:///opt/cactus/etc/amc13/
   AMC13XG_T2.xml</address-table>
    <fed-id>1376</fed-id>
</daqttc-mgr>
```

Last block of code in the hardware key is the ”connected-fed” tags. These tags define which output feds from the TWINMUX are connected with which input links in BMTF. Different feds are being propagated to different inputs. These lines are useful since they are necessary in order the auto-masking to work. Having the knowledge of which ports are related with which feds, one can implement a logic to mask them in the beginning of the run if a problem is reported to a specific fed.

```
<connected-fed id=”1390”>
    <port id=”wedge01.inputPorts.Rx[16:22]”/>
    <port id=”wedge02.inputPorts.Rx[16:22]”/>
    <port id=”wedge03.inputPorts.Rx[16:22]”/>
    <port id=”wedge04.inputPorts.Rx[16:22]”/>
    <port id=”wedge05.inputPorts.Rx[16:22]”/>
    <port id=”wedge06.inputPorts.Rx[16:22]”/>
    <port id=”wedge07.inputPorts.Rx[16:22]”/>
    <port id=”wedge08.inputPorts.Rx[16:22]”/>
    <port id=”wedge09.inputPorts.Rx[16:22]”/>
    <port id=”wedge10.inputPorts.Rx[16:22]”/>
    <port id=”wedge11.inputPorts.Rx[16:22]”/>
    <port id=”wedge12.inputPorts.Rx[16:22]”/>
</connected-fed>
```

Next is the Algo key. This key includes information which configure the algorithm. Two blocks of code exist, one with ”path-values” information for the registers of the algo block, and one more which is a mapping of the input links to the track segments and the wheel. Both blocks are presented below. Important lines are those which are highlighted. They configure the ”payload.mask_ctrl_N2” and ”payload.mask_ctrl_P2” registers. These line handle the masking of the inner DT stations at the wheels ±2. BMTF has different masking configurations for Collisions and Cosmics runs. In Collisions these stations masked since are being handled bu the OMTF system, in Cosmics these stations are unmasked.

```
<param id=”regTable” type=”table” delimiter=”|”>
    <columns>register_path| register_value</columns>
    <types>string| uint</types>
    <rows>
      <row>payload.run.run_param.open_lut | 0</row>
      <row>payload.run.run_param.sel_21 | 0</row>
      <row>payload.run.run_param.dis_newalgo | 0</row>
      ...
      <row>payload.mask_ctrl_N2 | 0x111111</row>
      <row>payload.mask_ctrl_P2 | 0x111111</row>
    </rows>
  </param>

  <param id="linkMapTable" type="table" delimiter="|">
    <columns>wheel_name | track_segment | track_segment_value</columns>
    <types>string | string | int</types>
    <rows>
      <row>N2 | left_ts1 | 0</row>
      <row>N2 | left_ts2 | 1</row>
      ...
      <row>00 | right_ts1 | 18</row>
      <row>00 | right_ts2 | 19</row>
      ...
      <row>P2 | left_ts1 | 32</row>
      <row>P2 | left_ts2 | 33</row>
      <row>P2 | right_ts1 | 34</row>
      <row>P2 | right_ts2 | 35</row>
    </rows>
  </param>
```

Infra keys are separated into two xml files, one for MP7s and one for AMC13s. These files hold the parameters which are required for the commands to run. For example, the last developed "BMTFReboot" command requires three parameters in order to run. The are being displayed in the code below, the first three parameters. For the AMC13s the commands are fewer that for MP7s, however the file folds the same structure.

```
  <infra id="bmtf">
    <context id="procMP7">
      <param cmd="BMTFReboot" id="fwToLoad" type="string">v2_4_0_9503_x4_zs_kmtf_v271.bin</param>
      <param cmd="BMTFReboot" id="algoRevToLoad" type="uint">0x95030160</param>
      <param cmd="BMTFReboot" id="fwRevToLoad" type="uint">0x12020400</param>
      <param cmd="reset" id="clockSource" type="string">external</param>
      <param cmd="reset" id="ttcConfig" type="string">external</param>
      <param cmd="reset" id="clockConfig" type="string">external</param>
      ...
      <param ns="bmtf" cmd="easyTxLatency" id="bankId" type="uint">0x2</param>
      <param ns="kmtf" cmd="easyTxLatency" id="bankId" type="uint">0x3</param>
      <param cmd="easyTxLatency" id="masterLatency" type="uint">350</param>
      <param ns="bmtf" cmd="easyTxLatency" id="algoLatency" type="uint">40</param>
      <param ns="kmtf" cmd="easyTxLatency" id="algoLatency" type="uint">56</param>
      ...
      <!-- RO Menu mode 1 capture 1 -->
      <param cmd="roLoadMenu" id="mode1:capture1:bankId" type="uint">0x2</param>
      <param cmd="roLoadMenu" id="mode1:capture1:delay" type="uint">0</param>
      <param cmd="roLoadMenu" id="mode1:capture1:enable" type="bool">true</param>
      <param cmd="roLoadMenu" id="mode1:capture1:id" type="uint">0x2</param>
      <param cmd="roLoadMenu" id="mode1:capture1:length" type="uint">5</param>
      <param cmd="roLoadMenu" id="mode1:capture1:readoutLength" type="uint">30</param>
      ...
      <param cmd="zsSetup" id="enableZS" type="bool">True</param>
      ...
      <!-- outputs : mask pT -->
      <param cmd="zsLoadMenu" id="capId[2]:enable" type="bool">True</param>
      <param cmd="zsLoadMenu" id="capId[2]:invert" type="bool">False</param>
      <param cmd="zsLoadMenu" id="capId[2]:mask" type="vector:uint">0x1ff, 0x0, 0x1ff, 0x0, 0x1ff, 0x0</param>
      ...
```

Last category is the Run Settings keys. These keys are dedicated in the masking of several components of a board. For example, a part of the AMC13 Run Settings key follows, it includes all the related MP7 boards to the top crate (each crate is being handled by one AMC13). The components are ready to be uncommented-out if needed.

```
<!--  <context id="AMC13_BOTTOM">
```

```
          <mask id="amcports.amc01" />
   <mask id="amcports.amc03" />
   <mask id="amcports.amc05" />
   <mask id="amcports.amc07" />
   <mask id="amcports.amc09" />
   <mask id="amcports.amc11" />
        </context> -->
```

The same logic also for the following code. This is the Run Settings key for the MP7s.

```
<!--  <context id="wedge02">
          <mask id="inputPorts.Rx01" />
   <mask id="inputPorts.Rx02" />
   <mask id="inputPorts.Rx03" />
   <mask id="inputPorts.Rx08" />
   <mask id="inputPorts.Rx09" />
   <mask id="inputPorts.Rx10" />
   <mask id="inputPorts.Rx11" />
   <mask id="inputPorts.Rx16" />
   <mask id="inputPorts.Rx17" />
   <mask id="inputPorts.Rx18" />
   <mask id="inputPorts.Rx19" />
   <mask id="inputPorts.Rx24" />
   <mask id="inputPorts.Rx25" />
   <mask id="inputPorts.Rx26" />
   <mask id="inputPorts.Rx27" />
   <mask id="inputPorts.Rx32" />
   <mask id="inputPorts.Rx33" />
   <mask id="inputPorts.Rx34" />
   <mask id="inputPorts.Rx35" />
        </context> -->
```

Appendix C

Validation Plots

The appendix presents the latest validation plots for the two algorithms which run in parallel in the BMTF FPGAs. Firstly plots for the hardware variables of the Phase-1 algorithm are displayed and following the old algorithm there are plots for Kalman algorithm too. For each BMTF algorithm validation for both Cosimics and Collisions modes are presented. The runs used are 321963 (cosmics) and 321933 (pp) for BMTF , while for Kalman the runs 324738 (Cosmics) and 325113 (pp).

For the BMTF the hardware variables that are being compared are ϕ, η, P_T, and the comparison of the reconstructed number of muons between the firmware and the emulator. In the plots the red histograms are emulated variables and the black marks are the firmware variables. Below each 1D plot the ratio of the two variables per bin is calculated.

For the Kalman algorithm validation of five variables is presented. Except ϕ, η, P_T and the 2D comparison regarding the number of reconstructed muons, plots for the hardware variables P_{T2} and d_{xy} are given too. The variable P_{T2} is the second measurement of reconstructed muon's momentum. This measurement (as is described in section 3.3) is a vertex unconstrained measurement suitable for studies related to displaced muons. The d_{xy} variable is the impact parameter of the muon encoded in 2-bits whose combinations specify ranged in cm.

Figure C.1: Cosmics validation plots for variables ϕ and η, Phase-1 Algorithm

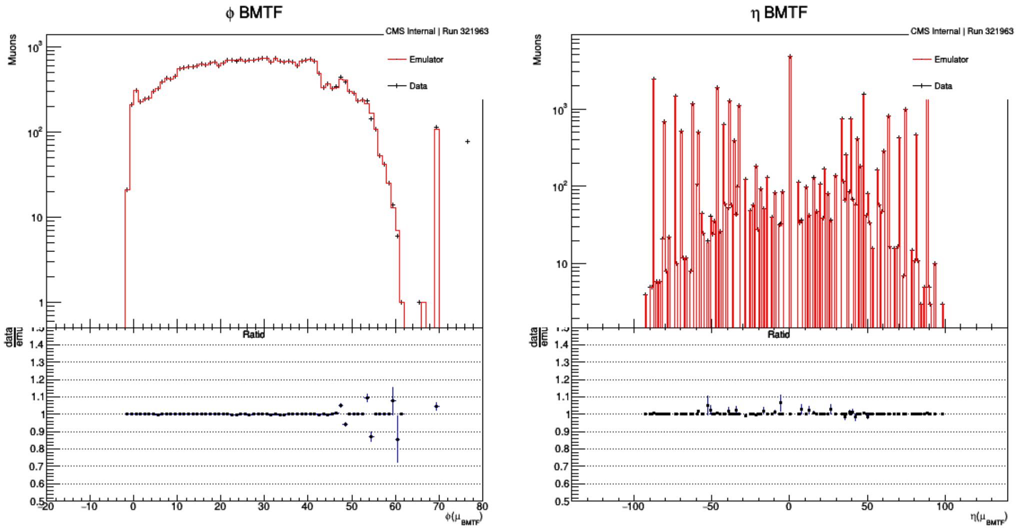


Figure C.2: Cosmics Validation plots for variables P_T and # of muons, Phase-1 Algorithm

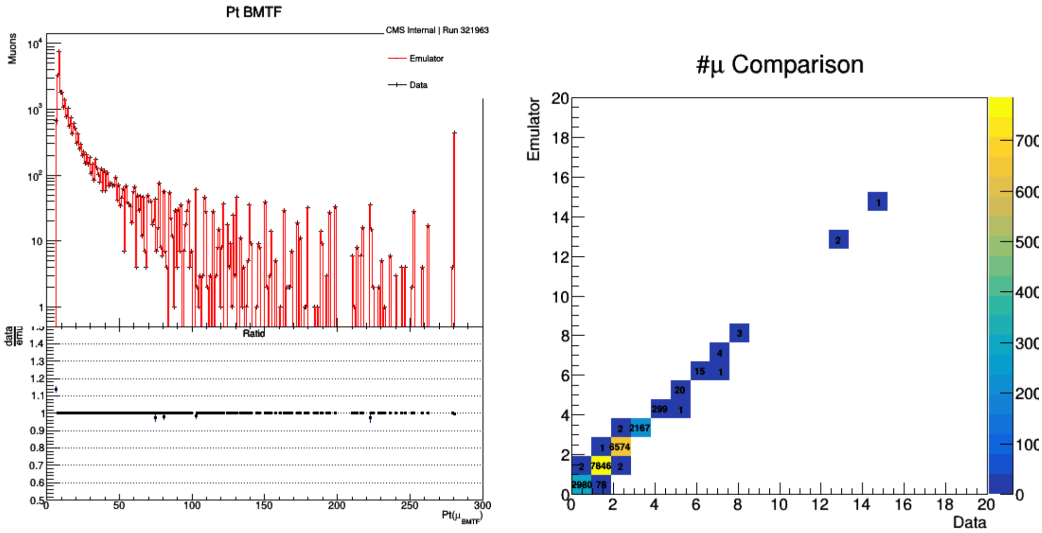


Figure C.3: Collisions validation plots for variables ϕ and η, Phase-1 Algorithm

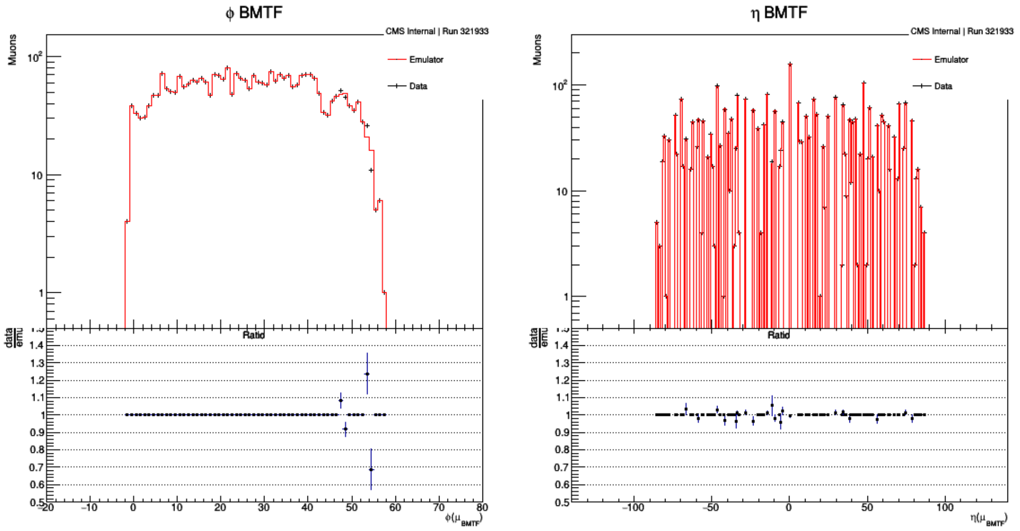


Figure C.4: Colisions Validation plots for variables P_T and # of muons, Phase-1 Algorithm

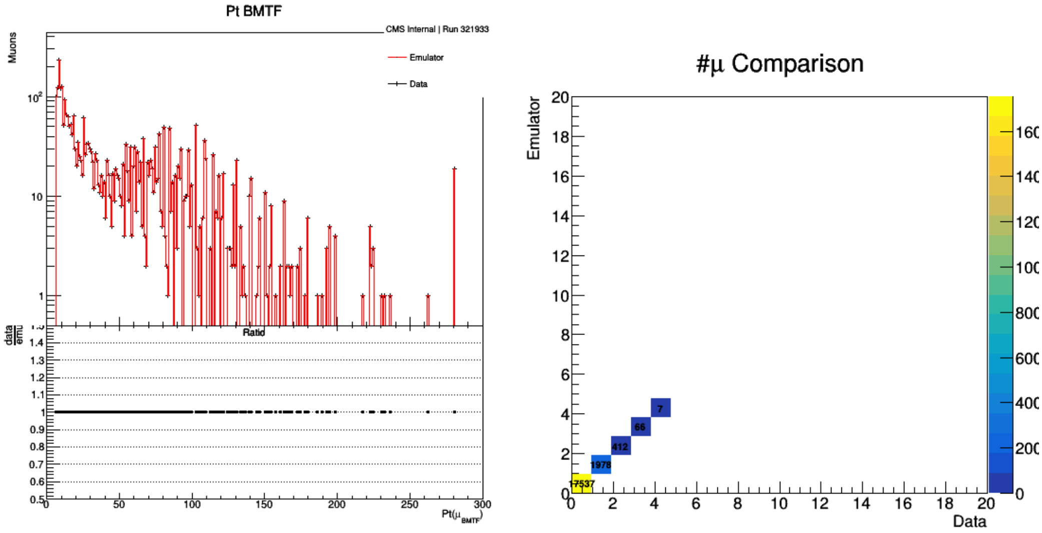


Figure C.5: Cosmics Validation plots for variables ϕ and η, Kalman Algorithm

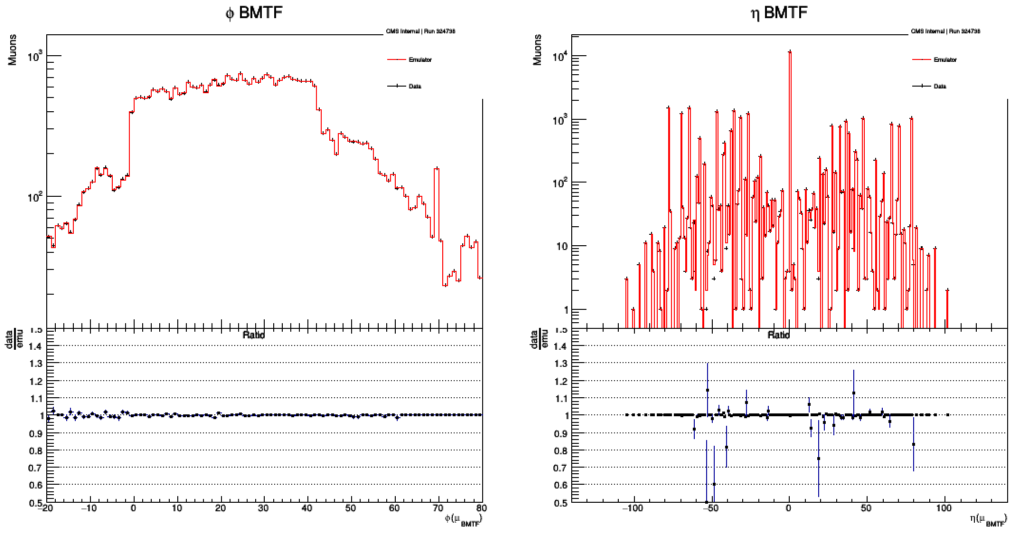


Figure C.6: Cosmics Validation plots for variables P_T and # of muons, Kalman Algorithm

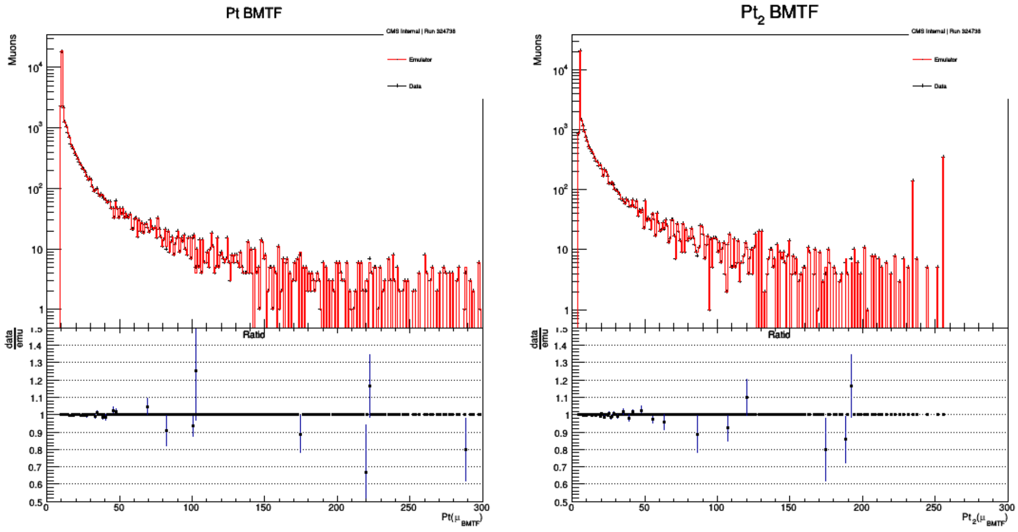


Figure C.7: Cosmics Validation plots for variables P_{T2} and d_{xy}, Kalman Algorithm

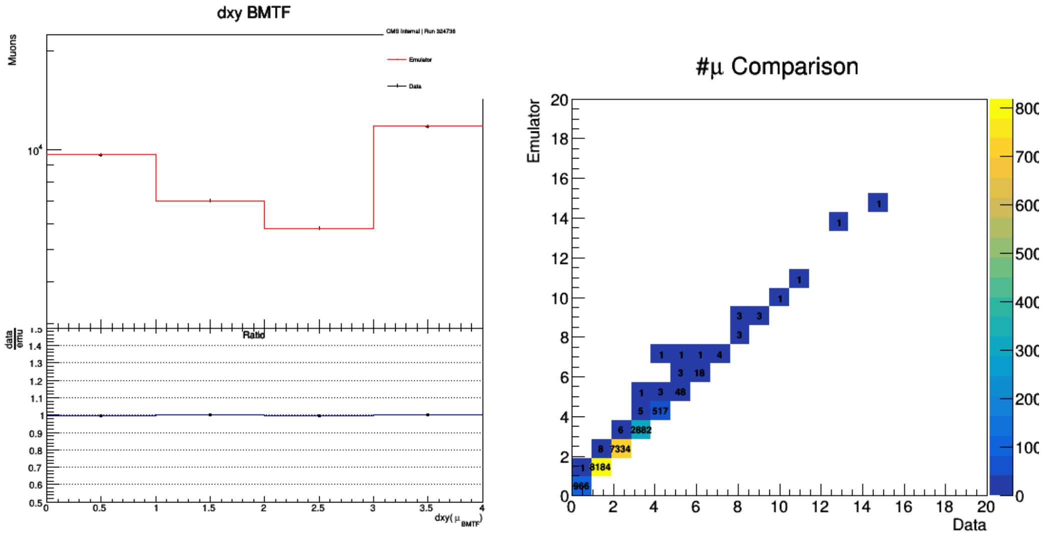


Figure C.8: Collisions Validation plots for variables ϕ and η, Kalman Algorithm

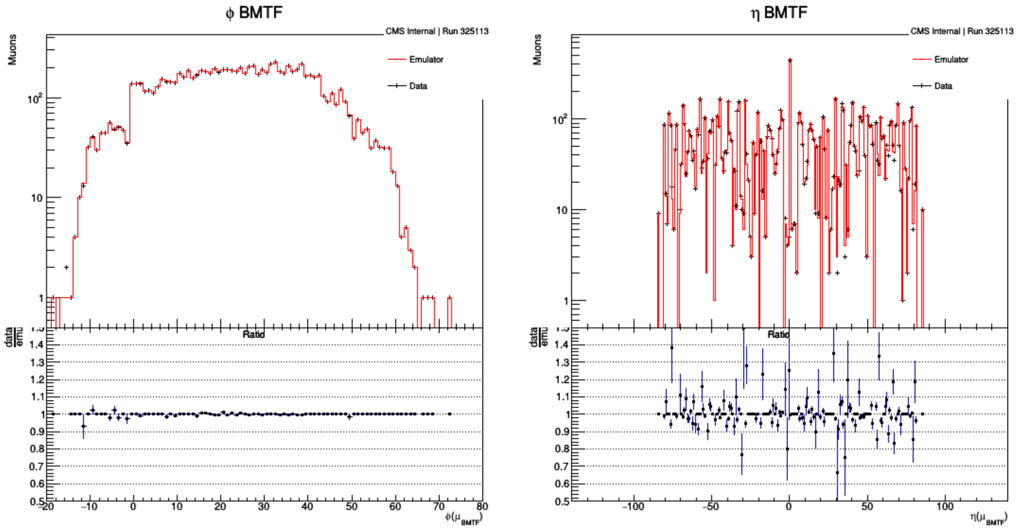


Figure C.9: Collisions Validation plots for variables P_T and # of muons, Kalman Algorithm

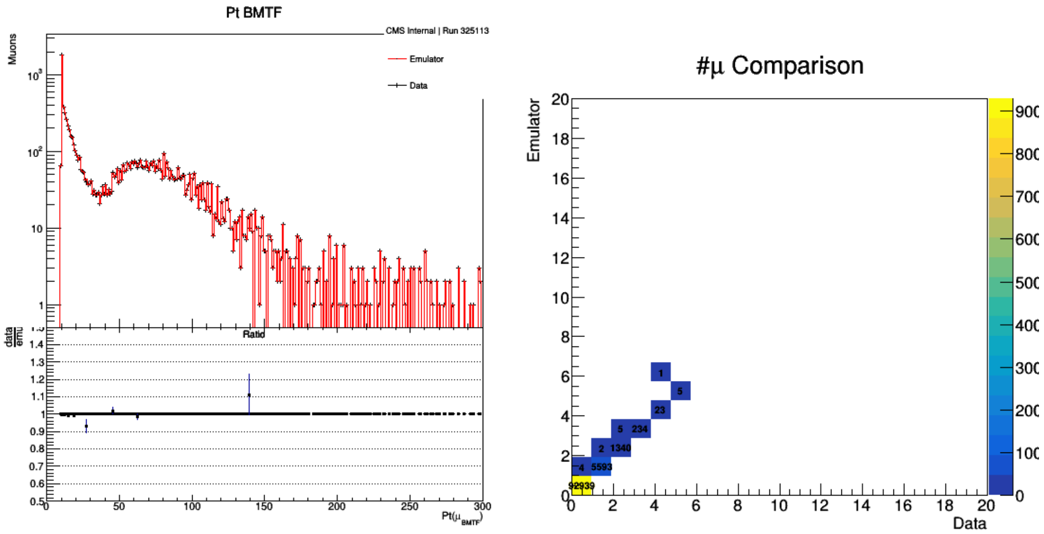
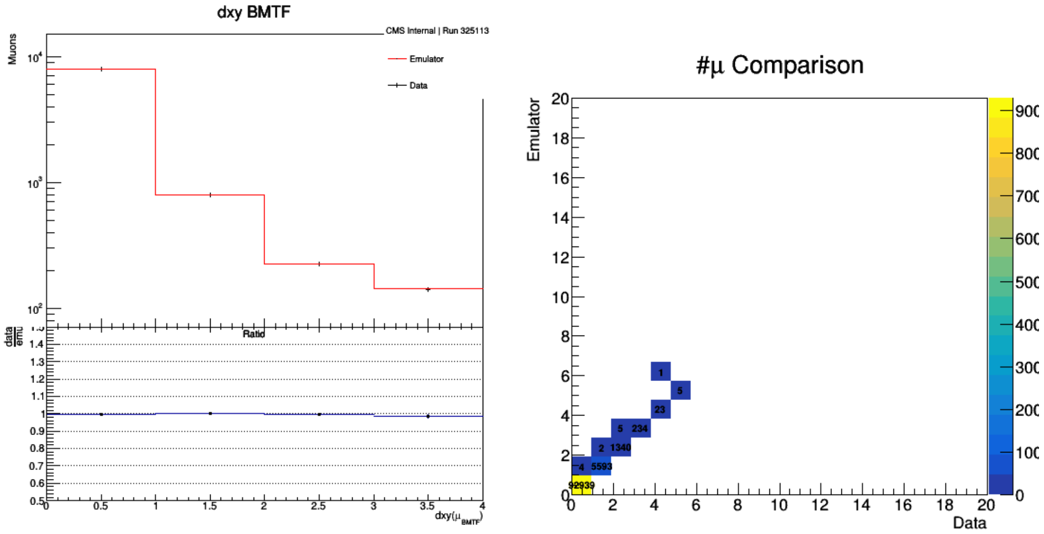


Figure C.10: Collisions Validation plots for variables P_{T2} and d_{xy}, Kalman Algorithm



List of Figures

Bibliography

[1] Quantum Field Theory. Mark Sredniki (2007)

[2] Quarks and Leptons: An Introductory Course in Modern Patricle Physics. Francis Halzen, Alan D. Martin (1984)

[3] CMS The Muon Project Technical Design Report. CMS Collaboration (1997)

[4] CMS The Electromagnetic Calorimeter Project Technical Design Report. CMS Collaboration (1997)

[5] CMS The Hadron Calorimeter Project Technical Design Report. CMS Collaboration (1997)

[6] CMS The Trigger and Data Aquisition Project, Volume I: The Level-1 Trigger. CMS Collaboration (2000)

[7] CMS Technical Design Report for the Level-1 Trigger Upgrade. CMS Collaboration (2013)

[8] Tony Lacey, ”Tutorial: The Kalman Filter”
http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman%20filter.pdf

[9] CMS Level-1 trigger online software team, ”SWATCH’s Documentation”
http://cactus.web.cern.ch/cactus/release/swatch/latest_doc/html/index.html

[10] CMS Twiki, ”CMSSW Application Framework”
https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookCMSSWFramework

[11] CMS Twiki, ”Recommendations for the 13 TeV data analysis, Run2016 (80X)”
https://twiki.cern.ch/twiki/bin/view/CMS/JetID13TeVRun2016

[12] CMS Twiki, ”Preliminary Recommendations for the 13 TeV data analysis, Run2017 (94X) Re-Reco”
https://twiki.cern.ch/twiki/bin/view/CMS/JetID13TeVRun2017

[13] LHC Physics Web, ”NLO-NLL hino-like chargino-neutralino (N2C1) cross sections”
https://twiki.cern.ch/twiki/bin/view/LHCPhysics/SUSYCrossSections13TeVn2x1hino

[14] C. Botta, R. Castello, G. Karathanasis, M. Peruzzi, G. Petrucciani, B. Schneider, P. Sphicas and M. Stoye 3, ”Search for new physics in events with two soft oppositely charged leptons and missing transverse momentum in proton-proton collisions at $\sqrt{s} = 13$ TeV”, *Phys. Lett. B*, doi:10.1016/j.physletb.2018.05.062